The dynamics of collective action in free and open source projects

Dissertation
for the Faculty of Economics, Business Administration and Information Technology of the University of Zurich

to achieve the title of
Doctor of Economics

presented by
Bernhard Kuster
from Uznach (SG) and Eschenbach (SG)

approved at the request of
Prof. Dr. Dr. h.c. Margit Osterloh
Prof. Dr. Egon Franck

The dynamics of collective action in free and open source projects

by Bernhard Kuster

First edition. Some rights reserved.

Survey reprint: University seal, FOSS logos, and typeface

The layout of the conducted survey (see Appendix I) has been supplemented with the seal of the University of Zurich and with logos of different FOSS projects. Due to the fact that the author is neither the owner of the corresponding intellectual property rights nor that rights are granted to put the emblems under a Creative Commons respectively a GNU FDL license these emblems have been removed. The typeface used in the survey is not protected by Swiss copyright law (see Blank 1999: 29). In consequence, the survey is reprinted with the original typeface.

Published by Bernhard Kuster

ISBN 978-3-033-01379-7

90000 >

9 783033 013797

ISBN 978-3-033-01379-7

The Faculty of Economics, Business Administration and Information Technology of the University of Zurich hereby authorises the printing of this Doctoral Thesis, without thereby giving any opinion on the views contained therein.

Zurich, April 1, 2009

the Dean: Prof. Dr. Dr. Josef Falkinger

“Satisfying developers is probably the biggest challenge to the open-source development model, one which no amount of technology or even money can really address. Each developer has to feel like they are making a positive contribution to the project, that their concerns are being addressed, their comments on architecture and design questions acknowledged and respected, and their code efforts rewarded with integration into the distribution or a really good reason why not.”

Brian Behlendorf (1999: 161), founding member of the Apache Group and former primary developer of the Apache HTTP server

Abstract

This dissertation deals with the dynamics of collective action in **f**ree and **o**pen **s**ource **s**oftware (**FOSS**) projects. Thereby the following questions are primarily answered: i) Is the problem of collective action shaped by dynamical forces? ii) How are different types of motivation related to the dynamics of collective action?

It is basically hypothesized that different development phases provide distinct incentives, opportunities, and limitations for contribution to FOSS. It is therefore assumed that developers with different types of motivation are differently attracted by diverse development phases. The relationship between development phases and motivation is *firstly* modulated by network externalities and switching costs respectively by the S-shaped production function of software. *Secondly*, the relationship is heavily affected by incremental decision making that changes payoffs for the developers. It improves the information of the successor and allows antecessors to make a credible commitment and fosters conditional cooperation.

The conducted empirical study supports the basic assumption that there is a relationship between development phases and motivation. Based on the findings of this dissertation one may presume that i) collective action is, depending on the production function of a collective good, to a greater or lesser extent shaped by dynamical forces and that ii) different types of motivation are related to the dynamics of collective action although there is no fix or universal relationship for all types of collective action.

As hypothesized the motivation related to human capital increases in the initial phase, decreases in the intermediate phase, and increases in the mature phase of FOSS development. The relationship between peer recognition and FOSS development is, as suggested, u-shaped. Investments in reputation increases with FOSS development going on not logarithmically as suggested but linearly.

However, not all predictions concerning the relationship between motivation and development phases proved to be correct (or almost correct). Contrary to the hypothesis pro social motivation increases about linearly during development. Also not in line with the hypothesis personal requirement answers to human capital.

Fun seeking was wrongly thought to decrease linearly in FOSS development because autonomy was predicted to vanish linearly. The data shows, however, clearly that fun seeking remains almost constant. The reason for this rather surprising finding may be that autonomy remains also more or less stable in all development phases. This result may be connected to the self-selection mechanism found in FOSS development. Because fun seeking is the most important type of motivation in FOSS development this has far reaching implications. Commercial projects may have a hard time to reproduce self-selection mechanisms. Employed programmers should select tasks to accomplish that are in the interest of the employer. In consequence, commercial projects that intend taking advantage of the FOSS production model should either deliberately foster other motivation types than fun seeking or treat the autonomy of their employees with special care.

Keywords: Collective action, free and open source software, development phases, motivation, governance structures and mechanisms, critical mass, tipping point, threshold models, incremental decision making, bandwagon effects, organizational life cycles, group dynamics.

JEL Classification: C12, D70, D82, D85, D91, H41, L17, L86, M15, M52, O32, Y4

Kurzfassung

Diese Dissertation beschäftigt sich mit der Dynamik kollektiven Handelns in **F**ree und **O**pen **S**ource **S**oftware (**FOSS**) Projekten. Dabei soll erörtert werden, ob und wie i) kollektives Handeln durch eine Dynamik beeinflusst wird und ii) welchen Einfluss dabei die Motivation der Programmierer hat.

Die Grundhypothese vermutet, dass unterschiedliche FOSS Entwicklungsphasen mit unterschiedlichen Anreizen, Chancen und Schranken einhergehen. Es wird deshalb angenommen, dass Programmier mit unterschiedlichen idealtypischen Motivationsarten sich in den verschiedenen Entwicklungsphasen ungleich stark beteiligen. Das Verhältnis zwischen Entwicklungsphase und Motivation wird dabei *erstens* durch Netzwerkexternalitäten und Wechselkosten respektive der S-förmigen Produktionsfunktion von Software bestimmt. *Zweitens* ist dieses Verhältnis beeinflusst durch inkrementale Entscheidungsfindungsmechanismen. Diese verbessern einerseits die Information für spät Entscheidende, andererseits können früh Entscheidende eine glaubwürdige Verpflichtung eingehen, was die konditionale Kooperationsbereitschaft fördert.

Die empirischen Resultate stützen die Grundhypothese eines Zusammenhanges zwischen Entwicklungsphase und Motivation. Es kann vermutet werden, dass i) kollektives Handeln, je nach Produktionsfunktion eines öffentlichen Gutes stärker oder schwächer, durch dynamische Kräfte beeinflusst wird und dass ii) verschiedene Motivationstypen unterschiedlich mit der Dynamik kollektiven Handelns verbunden sind. Dabei sind diese Verbindungen aber nicht generisch, sondern abhängig vom konkreten kollektiven Gut.

Wie vermutet steigt die Motivation von Investoren in Humankapital in einer initialen Phase an, dann sinkt sie ab um danach wieder zu steigen. Das Verhältnis zwischen Softwareentwicklung und den durch soziale Zusammengehörigkeit angezogenen Entwicklern ist, wie erwartet, u-förmig. Die Motivation von Investoren in Reputation steigt während der Entwicklung konstant an, indes linear und nicht wie vermutet logarithmisch.

Entgegen der Hypothese steigt jedoch die Motivation von pro-sozial motivierten Programmieren während der Entwicklung linear an. Die Motivation von Programmierern mit Interesse an einer bestimmten Funktionalität entwickelt sich unvermutet analog zu derjenigen von Investoren in Humankapital.

Es wurde vermutet, dass sich die Autonomie der Programmierer während der FOSS Entwicklung linear zurückentwickelt und dass daher gleichzeitig die Motivation spassorientierter Programmiere linear zurückgeht. Die empirischen Resultate legen hingegen nahe, dass diese Motivation konstant gleich bedeutend bleibt. Ein Grund für dieses eher überraschende Ergebnis könnte sein, dass die Autonomie der Programmierer in allen Entwicklungsphasen gleich hoch bleibt. Dies wiederum könnte auf ein Selbstselektionsmechanismus zurückzuführen sein. Dies hätte aufgrund der hohen Bedeutung von Spass bei der FOSS-Entwicklung weitreichende Konsequenzen. Kommerzielle Projekte können Selbstselektionsmechanismen kaum reproduzieren. Angestellte Programmierer sollen normalerweise Aufgaben bewältigen, die im direkten Interesse des Arbeitgebers liegen. Dies beschränkt die Möglichkeit, Selbstselektion im Rahmen kommerzieller Projekte einzusetzen. Wollen kommerzielle Projekte vom FOSS Produktionsmodell profitieren, so müssen diese andere Motivationstypen speziell fördern oder aber die Autonomie auf eine andere Art sicherstellen.

Schlagwörter: Kollektives Handeln, Free und Open Source Software, Entwicklungsphasen, Motivation, Organisationsstrukturen und -mechanismen, kritische Masse, Tipping Point, Schwellenwertmodelle, inkrementale Entscheidungsfindung, Bandwagoneffekte, organisationale Lebenszyklen, Gruppendynamik.

JEL Klassifikation: C12, D70, D82, D85, D91, H41, L17, L86, M15, M52, O32, Y4

Credit list

Certainly, no bigger project can be accomplished completely alone. This does also apply to FOSS development in which project developers are given credits for their input. Similarly, I would like to thank the following individuals:

- My parents for enabling me to study and for supporting me.

- Roger Luthi for giving a deep insight into the FOSS world and taking part in discussion in office 209. I am deeply indebted for his information what happens in the IT world. His bug reports prevented some considerable mistakes in the survey. The advices I was skeptical about turned out to be true.
- My friends for bothering and asking sometimes but not too frequently about the progress of this dissertation.
- Katja Rost for giving comments on the beta version of the survey and for explaining the difference between formative and reflexive constructs.
- Professor Margit Osterloh for being my Doctoral advisor and giving a deep insight into organization theory.
- The members and former members of the Chair for Organization Theory, Technology and Innovation Management at the University of Zurich for making coffee breaks and discussing various subjects.
- Prof. Dr. Egon Franck for being a reviewer of this dissertation.
- The participants of my survey as well as the organizers of the Chemnitzer Linuxtage and the FOSDEM conference.
- The FOSS round table at the ETH Zurich (Switzerland) for commenting a beta version of the survey.
- The authors of the WIDI study for providing their raw data.
- The Wikipedia community for providing a tremendous amount of information and being a valuable source of inspiration. Frequently, Wikipedia has been a starting point for a further inquiry. Because of its function as an initial promoter of an idea Wikipedia is probably underrepresented in the bibliography. The corresponding bias is hereby diminished.

Editorial note

The used typeface is Charis SIL. It is a serif typeface released under a free copyleft license (FSF 2006e). For the book cover the typeface FreeSans, a sans-serif typeface licensed under the GNU GPL, has been used.

List of abbreviations[1]

AIC	*Akaike Information Criterion*
AICc	*Akaike Information Criterion corrected*
ANOVA	*Analysis of Variance*
BIC	*Bayesian Information Criterion*
BSD	*Berkeley Software Distribution*
CEO	*Chief Executive Officer*
CSS	*Cascading Style Sheets*
CTO	*Chief Technical Officer*
CVS	*Concurrent Versions System*
DK	*Denmark*
DOD	*U.S. Department of Defense*
EPA	*Environmental Protection Agency*
FAQ	*Frequently Asked Questions*
FDA	*Food and Drug Administration*
FOSS	*Free and Open Source Software*
FSF	*Free Software Foundation*
FTC	*Federal Trade Commission.*
GNU FDL	*GNU Free Documentation License*
GNU GPL	*GNU General Public License*
GNU LGPL	*GNU Lesser General Public License*

[1] The abbreviations are based on the International Abbreviations and Acronyms Database (Thomson 2004), on usage by the corresponding organizations, or on accepted usage. Countries are abbreviated according to the two letter country codes of the ISO 3166-1 standard (ISO 2007).

HTTP	*Hypertext Transfer Protocol*
IAS	*International Accounting Standards*
ICT	*Information and Communications Technology*
IETF	*Internet Engineering Task Force*
IRC	*Internet Relay Chat*
JP	*Japan*
LAMP	*Linux, Apache, MySQL, and PHP (or Perl / Python)*
LEO	*Lyons Electronic Office*
MPL	*Mozilla Public License*
Ms-PL	*Microsoft Permissive License*
NL	*Netherlands*
NPL	*Netscape Public License*
NSNX	*Neither Selfish nor Exploited*
OEM	*Original Equipment Manufacturer*
OSI	*Open Source Initiative*
OSS	*Open Source Software*
R&D	*Research and Development*
SED	*Socialist Unity Party of Germany*
SU	*Union of Socialist Soviet Republics*
TIC	*Takeuchi Information Criterion*
UK	*United Kingdom of Great Britain and Northern Ireland*
US	*United States of America*
US-GAAP	*US Generally Accepted Accounting Principles*
ZAIG	*Zentrale Auswertungs- und Informationsgruppe*

Table of contents

Table of figures

1 Introduction

At October 6, 1989 the German Democratic Republic celebrated its 40th anniversary. The general secretary of the SED Erich Honecker called the state foundation a historical necessity. Very shortly afterwards Honecker resigned, the Berlin Wall fell, and the whole state ceased to exist. What happened? These events were prominently driven by the dynamics of mass protests or with other words by the dynamics of collective action (see chapter 6.4).[2] The protests in Leipzig, known as the Monday demonstrations, were the most important mass protests. Starting in September 1989 the number of protestors increased from a few thousands to almost half a million in November. With more demonstrators joining collective action the probability of success increased while at the same time the probability and severity of sanctions decreased. In consequence, even more opponents joined the cause that in turn attracted additional demonstrators. By increasing their size the protestors boosted the pressure on the regime considerably. As a result of this escalating success of collective action the breakup of the German Democratic Republic became manifest.

The dynamics of the mass protests described above seems to have been related to a change in the type and intensity of motivation of the protestors. For example it can be presumed that the motivation to oppose the regime of the first demonstrators was so intense that they accepted major costs for doing so. They wanted "to live within the truth" as Václav Havel (1985: 39) prominently called it. The costs of protest decreased considerably with an increased number of protestors. In consequence, protestors with a less intense motivation found it worthwhile to join collective action in a later phase.

As shortly discussed above incremental decision making played a major role in the collapse of the German Democratic Republic. The decision of a few individuals to join the protests induced even more individuals to do so as well. That collective action is distinctively shaped by incremental decision making is not isolated to the collapse of the German Democratic Republic. Imagine as another example a step-level public good such as the lighthouse (see chapter 5.7). The decision situation of individual A who lays the very first brick of a new lighthouse is very different from the decision situation of individual Z who inserts finally the light bulb. Individual A has to accept the risk that he will be the only one who contributes and that the lighthouse will never be functional.[3] Individual Z knows in contrast for sure that after his contribution the lighthouse will be functional. That the decision situation of contributions to public goods changes may have important consequences. It can be assumed that the preference of individual A concerning the construction of the lighthouse is probably higher than the one of individual Z.[4] The dynamics of collective action seems therefore

[2] Other important factors were the success of the opposition in the semi free elections in Poland, the announcement that the Soviet Union shifted from the Brezhnev Doctrine to the Sinatra Doctrine, and the mass migration to Austria.

[3] Certainly, individual A may have different strategies to reduce the risk to be the only one who contributes. He may for example make a contract with individuals B to Z. However, one may imagine situations in which it is practically not possible or too costly to reduce the discussed risk noticeably.

[4] It may be that individual Z has the higher motivation than individual A. In such a case individual Z had perhaps no chance to contribute earlier. Individual Z may also be very poor so that he had to save money before he was able to pay for the light bulb. However, if one imagines an indefinite number of corresponding examples one may assume that individual A has on average the higher motivation than individual Z.

to be related to the motivation of individuals. This in turn is relevant for somebody organizing collective action. A promoter of the lighthouse should probably focus on the first contributors, let us say individuals A to E. Convincing these individuals to contribute if success is uncertain is difficult. Doing so if success is close seems to be in many cases easier. Individuals V to Z see that their contribution will put the lighthouse into function. They hardly need much incitement.

Based on the brief comments made above it seems to be evident that in many cases of collective action its dynamics plays an important role. Certainly, the dynamics of collective action related to the downfall of the German Democratic Republic is not the same as for the construction of a lighthouse. The object of collective action matters. The object of this study, namely **F**ree and **o**pen **s**ource **s**oftware (**FOSS**), is a very special public good. This dissertation attempts to discuss the dynamics of collective action in FOSS projects. As a byproduct of this analysis some insight concerning the dynamics of collective action in general is anticipated.

FOSS projects are interesting objects for research beyond software development. *Firstly*, many developers are not paid for their contributions. The reasons why they participate nevertheless are multifaceted. Because of this FOSS is an interesting example how differently motivated individuals can work together. *Secondly*, FOSS development is based on strange governance structures and mechanisms that provide restrictions and incentives for collective action. In this respect user involvement and peer production can be exemplary mentioned. Knowing more about such structures and mechanisms is interesting because there is some indication that they get more and more important in economic life. *Thirdly*, FOSS is quite successful. Analyzing the development of FOSS may produce knowledge how the success of the FOSS innovation model can be adopted beyond software development.

In the following the theoretical frame, the object of the study, and the basic research questions are briefly introduced. Afterwards, the relationship to existing FOSS research, the further procedure of this dissertation, and some reservations are discussed.

1.1 The theoretical frame

The dynamics of collective action has hardly attracted the attention of academic research (however see Heckathorn 1996). Mostly related concepts have contributed to the understanding of the dynamics of collective action. Examples are critical mass theory (Marwell & Oliver 1993; Oliver, Marwell, & Teixeira 1985; Schelling 1978), research about tipping points (Granovetter 1978; Grodzins 1958; Schelling 1971a), threshold models (Granovetter 1978; Wood & Doan 2003; Yin 1998), and bandwagon effects (Leibenstein 1950).

There is almost no research about collective action in general respectively its dynamics that includes different types and intensities of motivation. A notable exception is Elster (1989a: 187; 1989b: 131) who, very briefly, connected different types of motivation with the dynamics of collective action (see chapter 6.3).

Several questions arise that are important for the further course of this dissertation. *Firstly*, why is it important to know how to solve the problem of collective action? *Secondly*, how does incremental decision making influences collective action? *Thirdly*, why is motivation relevant if the dynamics of collective action is analyzed? In the following these questions are briefly discussed in turn.

Why is it relevant to solve the problem of collective action?

In the resource based view a sustainable competitive advantage rests on an optimal mix of physical, organizational, and human resources (Barney 1991; Prahalad & Hamel 1990). To be a source of sustained competitive advantage such resources must be valuable, rare, reasonably durable, and difficult to imitate or substitute (Barney 1991).

Organizational knowledge in the form of encultured and encoded knowledge (Blackler 1995: 1024-1026; Collins 1993: 96-99) meets the above mentioned requirements for valued resources (see also Lado & Wilson 1994; Spender 1996: 46). Organizational knowledge is based on the history of interactions in an organization. It is in consequence path dependent and therefore rare and unique. There are no two organizations that have the same history of learning experiences (Cabrera & Cabrera 2002). Other organizations can hardly appropriate organizational knowledge because of its supra-individual character. Organizational knowledge differs from individual forms of knowledge insofar as it is not held by one person but that it is embedded in shared understandings (encultured knowledge) and in abstract and shared symbols (encoded knowledge). It is thereby based on interactions among individuals. Organizational knowledge is difficult to imitate because of causal ambiguity; organizations are a network of formal and informal interpersonal relationships and have a system of norms and beliefs (Sanchez & Heene 1997). Sustainable competitive advantage based on organizational knowledge depends in consequence not on outstanding individuals but on the collaboration and the interaction of a group of outstanding individuals (Wright, Dunford, & Snell 2001).

If the above mentioned collaboration works well and the individuals cooperate with each other a cooperation rent may be the result. However, the production of such cooperation rents is tied with the problem that for an individual benefits and costs of cooperation are not coupled directly. In consequence, a social dilemma may arise in that i) each individual receives a higher payoff for a socially defecting choice than for a socially cooperative choice and in that ii) all individuals are better off if all cooperate than if all defect (Dawes 1980: 169). Miller (1992: 35) argues that the management of social dilemmas and thereby the problem of collective action is at the heart of the managerial problem. The solution to pay for cooperative behavior is not easy to implement. It is not trivial to determine i) what is cooperative behavior and ii) how it is recognized by the management. Spender and Grant note (1996: 8):

> *The dilemma for management is that, for the same reasons that competitors cannot replicate the firm's knowledge, so the firm itself may not understand it well enough to exploit it effectively.*

In consequence, researching collective action is not trivial but does nevertheless provide crucial information for the successful management of organizations.

How does incremental decision making influence collective action?

The decision situation an individual usually faces is very different from a Crusoe economy (Von Neumann & Morgenstern 1947: 10) in that the actions of others are not relevant. Solving[5] social dilemmas depends in many cases on sequential chains of cooperation (Macy 1991). Individual choices to contribute to collective

[5] It is not taken for granted that solving social dilemmas is always desirable for a larger community (see chapter 1.6). Cooperation within a group may lead to monopolistic business cartelization or racial discrimination with adverse effects for outsiders (Cowen & Sutter 1999: 163).

action may be contingent on the cooperation of others.[6] In consequence, the dynamics of collective action is essentially based on incremental decision making. There are five main reasons why incremental decision making influences individual decision making:

Firstly, incremental decision making improves the *available information* for the successor. He knows what others did and may therefore adjust his decision. This is clearly visible in the stag hunt game.[7] The corresponding coordination dilemma vanishes completely if the actors are allowed to decide incrementally.

Secondly, incremental decision making allows antecessors to make a *credible commitment*. Making such a commitment is especially powerful if a decision cannot be reverted. This is impressively shown in the big monkey and little monkey game reported by Gintis (2000: 3-9).[8] In this game the actor moving first can force the other actor to invest scare resources by making the credible commitment of doing nothing. The power of making a credible commitment can be demonstrated by the decision of a general burning the bridge behind him. Doing so is a very credible commitment not to retreat. Similarly, in the Spanish conquest of Mexico Hernándo Cortés ordered to scuttle the ships to force his soldiers to fight seriously and to eliminate any means of desertion. Making heavy investments in technology or customers which are lost if a firm withdraw from business is also a very powerful commitment to stay in business.

Thirdly, *payoffs* related to cooperation may change contingent on the decisions of others. The change in payoffs may be based on a psychological desire to join a crowd, to be one of the boys, or to jump on a bandwagon (Leibenstein 1950: 184). Payoffs may also change because of network effects. In the case of positive network effects the utility that a user derives from the consumption of a good increases with the number of others consuming the good as well (Katz & Shapiro 1985). An example is the telephone. Its value increases with the number of other users.[9]

Fourthly, incremental decision making influences *conditional cooperation*. For Fehr and Fischbacher (2004: 186) human cooperation is largely based on a social norm of conditional cooperation. According to this norm one cooperates if others had cooperated previously. Corresponding experiments suggest that conditional cooperation is indeed important for cooperation (Brandts & Schram 2001; Fischbacher, Gachter, & Fehr 2001;

---

[6] It may be enough that individuals *believe* that others have taken a certain action to let them act (Merton 1948; Thomas & Thomas 1928).

[7] The stag hung game describes a dilemma in that two individuals go out hunting (for further information see chapter 5.4.2.1.1). For each hunter it is possible to go for a stag or a hare. To catch a stag it is necessary that both go for it. At the same time it is possible for a hunter to catch a hare without the help of the other. Though, a hare has less meat than a stag and is therefore less worth even if the stag has to be equally shared. Both hunters prefer therefore to go for the stag but only if they can be reasonable sure that the other hunter goes for the stag as well. The stag hunt game is in consequence a coordination game in that rational egoists prefer to cooperate with each other. The problem for successful cooperation is coordination respectively to learn about the choice of the other player.

[8] In the big and little monkey game an important part of the nutrition of two monkeys is the warifruit from the waritree (for further information see chapter 5.4.2.1.1). Warifruits have 10 kilocalories of energy. Waritrees produce sporadically one fruit at the very tip of the tree. To get the fruit at least one monkey must climb the tree. The cost for climbing the tree is 2 kilocalories for a big monkey and almost nothing for a little monkey. The little monkey is very skillful so that the fruit does not fall to the ground immediately if it is approached. The big monkey is impetuous so that the warifruit falls to the ground if he tries to catch it. The monkey that does not climb starts to eat the fruit if it falls to the ground. Because of this and because there are costs for climbing it is for both monkeys better if the other climbs. In this situation it is crucial which monkey is able to force the other to climb by making a credible commitment to do nothing. However, making a credible commitment is only possible if the game is played sequentially.

[9] Certainly, if network resources are limited negative network effects may arise if the number of users is above the congestion point. The effect of network congestion in IP telephony may be for example packet losses.

Keser & van Winden 2000).[10] Certainly, conditional cooperation is only successful if there is at least one actor who cooperates unconditionally and therefore induces the cooperation of others. If decisions are made sequentially and not incrementally there is of course nobody who has cooperated unconditionally before.

Fifthly, incremental decision making activates the *shadow of the future*. Thereby strategic interaction is encouraged. The tit for tat strategy often discussed in game theory (Axelrod & Hamilton 1981; Boyd & Richerson 1988; Nowak & Sigmund 1993) may ensure cooperation between rational egoists.

Based on the argumentation presented above it seems to be clear that incremental decision making is very important for discussing the problem of collective action.

Incremental decision making is also relevant beyond social dilemmas. Having the choice between two restaurants A and B not known to a certain individuals the previous decision of others provide crucial information. An individual standing in front of the two restaurants is able to see how many individuals sit in each restaurant. If restaurant A is fairly occupied and restaurant B completely empty most individuals probably choose the former one. The individual may assume that others have good information about the cookery in the two restaurants and have decided that A is the better one. Additionally, most individuals like to be in company. Eating alone in an empty restaurant is not fun. Such herding behavior can be observed in many areas. Individuals visit art exhibitions everybody else visit and if the alpha male in schools starts wearing a certain sneaker brand very soon others will follow until most wear the same brand.

Why is motivation relevant if the dynamics of collective action is analyzed?

Social dilemmas and the problem of collective action are related to the motivation and personality of the involved subjects. Basically, individuals with competitive or individualistic motives display on average higher non-cooperative behavior than the ones with altruistic or cooperative motives. If only saints maximizing the welfare of others are involved in the production of a public good many problems of collective action vanish. At the same time it gets tremendously difficult to produce a public good if only subjects with a sadistic personality disorder who have pleasure in the psychological or physical suffering of others are involved. Sen (1977: 335) argues that "to run an organization *entirely* on incentives to personal gain is pretty much a hopeless task".

It is argued that motivation does not only play a major role for collective action in general but specifically also for its dynamics. The interaction of incremental decision making and different types of motivation is very relevant for collective action respectively the production of public goods. For example the degree other individuals have already contributed to a public good influences the decision of an individual motivated by fairness respectively conditional cooperation. Individuals motivated by conditional cooperation norms will not start cooperation because their decision is contingent on the decision of others. In contrast to such individuals altruists will cooperate unconditionally. In consequence, such individuals may start collective action. This clearly shows that motivation interacts with incremental decision making.

[10] In the experiment of Fischbacher et al. (2001) participants had to fill out a contribution table in which they had to indicate their own contribution for each possible average group contribution. Therefore, the decision to contribute is not made incrementally but simultaneously. However, the decision situation is partly similar to incremental decision making because the behavior of an individual is contingent on the decision of other individuals. There is one but very important difference to real incremental decision making: emotional outbursts after noncooperation are ruled out (see also chapter 5.4.1).

The interaction of incremental decision making and motivation can be impressively pinpointed by the development of the number of participants in the Monday demonstrations already mentioned before. For almost a decade only few opponents of the regime participated in the prayers for peace that preceded the Monday demonstrations. It can be assumed that these individuals were strongly driven by what Václav Havel (1985: 39) called impressively “to live within the truth”. However, after a certain mass of demonstrators was reached in autumn 1989 the state lost practically the ability to punish the opponents. Thereby the costs of opposition decreased and individuals with other types of motivation or with a less intense motivation joined the demonstrations. As a result, the number of demonstrators increased sharply.

1.2 The object of the study

FOSS applications such as Linux[11], Apache, and Firefox have grown mature. They are more and more adopted by companies, organizations, and individuals (Feller & Fitzgerald 2002: 2-5, see also chapter 2.10). A study for the U.S. Department of Defense (DOD) concluded that a ban of “FOSS would have immediate, broad, and strongly negative impacts” on the DOD (MITRE 2003: 2). Such a ban would for example “result in at least a temporary shutdown of many or most of its web-based network and services” (MITRE 2003: 20).

FOSS applications are special public goods. FOSS licenses grant the right to study, change, improve, and redistribute its source code. These rights are granted to everybody irrelevant whether subjects have made any contributions or not. In consequence, social dilemmas and the problem of underprovision in collective action arise. The question how and why the FOSS innovation model succeeds is therefore not trivial.

Successful FOSS projects do not only depend on technical performance but also on the ability of the project leaders to form a viable developer community (Bezroukov 2004; Raymond 1999a: e.g. 36-38). Compared to other Unix-like[12] alternatives (FreeBSD, OpenBSD, or NetBSD) Linux was for a long time technically not superior. The advantage of Linux was its development model. Torvalds “was keeping his hacker/users constantly stimulated and rewarded - stimulated by the prospect of having an ego-satisfying piece of the action, rewarded by the sight of constant (even daily) improvement in their work” (Raymond 1999a: 40). It can therefore be assumed that satisfying the motivational needs of developers is one of the biggest challenges in the FOSS development model (see Behlendorf 1999: 161). Most developers are neither paid nor employed by the project they contribute to. In fact, there are many different reasons why developers are motivated to contribute (David, Waterman, & Arora 2003b; Ghosh 2005; Hars & Ou 2002; Hertel, Niedner, & Herrmann 2003; Lakhani & Wolf 2005; Mitsubishi Research Institute 2004c, 2004d; Roberts, Hann, & Slaughter 2006; von Krogh, Spaeth, & Lakhani 2003). Not all developers contribute to FOSS projects for the same reason. This results in interesting interactions between individuals with different types of motivation.

Large traditional software houses such as IBM not only use FOSS products but also adopt the FOSS innovation model (e.g. Eclipse project) or contribute to FOSS projects (e.g. Linux). The magnitude of firm involvement in FOSS can be anticipated by the highly remarked IBM announcement to spend more than 1 billion $ in 2001 for Linux development (Wilcox 2000). In 2002 IBM claimed to recouped most of this spending by sales of related software and systems (Shankland 2002). In 2006 IBM (2006) employed more than 300 full time

[11] For a short discussion of the GNU/Linux naming controversy see footnote 37.

[12] There are two different spellings of Unix: Unix and UNIX. For no other reason than following the spelling of Ritchie (e.g. 2001) only the first letter is capitalized.

engineers "on Linux as part of the open source community". This is very surprising because the FOSS governance structures and mechanisms are very special. Superficially, one might argue that these structures and mechanisms are not very suitable for business companies. Von Krogh and von Hippel (2006: 975) state concerning FOSS development: "Who would ever have imagined that innovation could flourish under conditions like those?". Such peculiar structures and mechanisms are for example peer review and peer production, self-selection as a recruitment principle, computer-mediated interaction, heavy user involvement, and modularization. That researching FOSS governance structures and mechanisms is rewarding is discussed on the basis of peer production and user involvement:

- FOSS development is based on ***peer production***. Peer production is a phenomenon with broader economic implications than thinking about FOSS alone would suggest (Benkler 2002: 444). According to the Wall Street Journal one of workplace's most vexing problems is "how to have employees collaborate and communicate better electronically" (Swisher 2004: B1). Discussing the ability of Toyota to innovate Hamel (2006: 74) argues that "Toyota's real advantage was its ability to harness the intellect of 'ordinary' employees".[13] In a noticed book Himanen (Himanen 2001: ix) suggested that FOSS development processes are the new work ethic challenging the Protestant work ethic (Weber 1992). Yuan et al. (2005: 213) observe that collective-oriented forms of organization and management grow in relevance.

  The driving forces behind the growing importance of peer production in general are the declining price of physical capital involved in information production, the declining price in communication, and the growing importance of human capital. The growing importance of human capital is associated with the problem of monitoring. Drucker (2001) argues that in a knowledge society in which knowledge travels even more effortlessly across organizational borders than money and everybody may easily acquire knowledge through formal education the problem of monitoring gets very difficult. If monitoring of employees gets more difficult problems such as shirking raises (Hansmann 1990: 1761). Alchian and Demsetz (1972: 786) argue that the ability to monitor depends on the easiness to relate input to output. Peers qualify better to monitor others in knowledge intensive areas than managers because they know better how to relate input to output. If the employees have additionally to bear the costs of shirking they will have an incentive to monitor their fellow workers and apply pressure to not shirk. That is why peer production is frequently found in law firms and other groups with artistic or professional skills (Alchian & Demsetz 1972: 790).

  However, despite the discussed strengths peer production is not without pitfalls. According to Benkler (2002: 378) there are primarily two problems associated with common based peer production:

  i. If no one can appropriate the benefits of investments, rational egoists will not invest scare resources. Rational egoists do not invest in something that "may be taken from him tomorrow" (Rose 1986: 712). In short, there is no *motivation* for rational egoists to contribute to a collective good (see chapter 3).[14]

[13] The example of Hamel's (2000: 211-222) appraisal of Enron's innovation style shows that even high-rated business scholars may be very wrong.

[14] For definitions of social dilemma, collective action, collective good, public good, and common good see chapter 5.2.

ii. If there is nobody with the power to organize collaboration it may fail. The problem of peer production may be that “no one knows whom to approach to make exchange” (Rose 1986: 712). There is therefore a problem of *coordination* (see chapter 4).

It can be summed up that peer production is an interesting but not a trivial manner of production. To illuminate related pitfalls corresponding research is needed. Peer production is also found in commercial firms. To research FOSS development may therefore give some interesting insight for product development within commercial firms.

- ***User involvement*** is a very important aspect of innovation (von Hippel 1988: 11-27). In fact, the world’s first business computer was developed by a catering and food-manufacturing company that needed computing power and not by a electronic or office equipment firm (Aris 2000; Caminer 2003; Hendry 1987; Land 2000). Today, user involvement seems not to have lost its importance. FOSS is probably one of the most thrilling examples of user involvement and user innovation. In a report the U.S. President's Information Technology Advisory Committee (PITAC 2003: 4) argued that it is interesting to study FOSS development because it is a unique mode of production with a mixture of public, private, and academic partnerships.

  User involvement is not restricted to computer hardware and software development. Von Hippel (2005a: 268-269) reports as an example for user innovation high-performance windsurfing. The extreme growth of Wikipedia (see chapter 4.4) is another up to date example of the power of user involvement. Yet another example of user involvement is citizen journalism respectively customer generated content. After catastrophes newspapers and television stations use more and more content such as images or videos that is provided by its customers. After the July 7, 2005 London bombings that hit London's public transport system BBC received from customers 300 images and several video sequences within the first hour (Douglas 2006). Concerning the blast at the Buncefield oil depot in Hertfordshire at December 11, 2005 BBC received the first image within 13 minutes after the explosion followed shortly by the first video.[15] Very soon BBC got 5,000 images and at the end of the day BBC had received 10’000 images. Some mass media have installed special programs designed to collect and utilize such customer generated content. CNN (2007) for example installed the I-Report program in which “video, a photo, a piece of audio or a text story” can be submitted. In the Virginia Tech school shooting at the April 16, 2007 the CNN news coverage was highly dependent on such a submitted video (Hanfeld 2007; Kaiser 2007).

  Because user involvement and user innovation seem to get more important in the future (von Hippel 2005a: 121)[16] the proper understanding of this phenomenon seems to be important.

A book for managers interested in starting a FOSS project advises “what to avoid” (Freiberger & Swaine 1984: 249-271). The first advice is to avoid “not understanding open source”. Indeed, one can hardly starting a FOSS

[15] The Buncefield oil depot blast is said to have been the largest in Europe during peacetime (BBC News 2005). Due to the enormous amount of energy and the weather conditions the blast seems to have been heard even in the Netherlands and Belgium.

[16] There are many drivers for this observation. Without going into details increased education, increased household income, and increased access to information by the internet may be some reasons why user involvement might get more important.

project without knowing the phenomenon. The same holds true for projects using the FOSS innovation model beyond software development. This dissertation attempts to contribute to the corresponding knowledge.

1.3 Research questions

At the very heart of this dissertation there are two connected sets of questions. *Firstly*, there are basic research questions related to the dynamics of collective action. These basic research questions are associated with a general problem. In consequence, they are not solely connected to FOSS development. This dissertation does not answer these research questions directly. They are addressed indirectly by analyzing the FOSS innovation model (see below). From the analysis of FOSS development some insights for the general problem of the dynamics of collective action are deduced.

Secondly, the very same basic research questions are adapted to FOSS development and the FOSS innovation model. These adapted research questions serve i) to find out more about the dynamics of collective action in general (see above) and ii) to analyze the FOSS innovation model itself.

In the following, the basic research questions are presented. Afterwards, these questions are adapted for FOSS development.

The basic research questions

The basic research questions are related to the dynamics of collective action in general:

i. *Is the problem of collective action really shaped by dynamical forces?*

   Although there is already some support that collective action is shaped by dynamical forces additional empirical research does further the understanding of the corresponding mechanisms. If FOSS development is really shaped by dynamical forces the basic idea is even more corroborated. Otherwise knowledge is created about the conditions in that collective action is not shaped by dynamical forces.

ii. *How are different types of motivation related to the dynamics of collective action?*

   As already mentioned there is almost no research dedicated to the question how the motivation of individuals is related to the dynamics of collective action. By discussing the corresponding question for FOSS development the assumption about the importance of motivation for the dynamics of collective action is supported in general.

iii. *How are governance structures and mechanisms related to the dynamics of collective action?*

   There is good knowledge about the relationship between governance structures respectively mechanisms and collective action. However, the hypothesis that these structures and mechanisms have to be adapted to the dynamics related to collective action is hardly acknowledged. The proper adaption of governance structures and mechanisms in the course of action is heavily related to the management of collective action. A promoter of collective action is interested in knowing which conditions boost the involvement of potential contributors. Research related to governance structures respectively mechanisms and collective action is therefore highly relevant for practice.

The basic research questions applied to FOSS

The basic research questions discussed above can be applied to FOSS development respectively the FOSS innovation model:

iv. Is the FOSS innovation model shaped by different development phases?

v. Is there a relationship between motivation and ongoing FOSS development?

vi. How are governance structures and mechanisms related to FOSS development phases?

As already mentioned answering these questions furthers the knowledge i) about the dynamics of collective action in general and ii) about the FOSS innovation model itself.

What can be additionally learned?

All readers not interested in the research questions presented above may find it nevertheless rewarding reading the dissertation because one can also learn the following:

- ***Zoologists*** can learn what happens if money is introduced to a community of tufted capuchin monkeys (chapter 1.5).
- ***Prospective Nobel Prize winners in physics*** can learn what it takes to be one (chapter 3.2.1).
- ***Political scientists*** can learn that during the time of absolute monarchs the pirate community in the free colony of Libertalia at Madagascar was one of the few true democracies (chapter 4.5). *Captains* can learn that the mutiny on the Bounty cannot be explained by captain Bligh's cruelty.
- ***Lion tamers*** can learn what happens according to Einstein if one forces a greedy beast of prey to devour constantly (chapter 5.6).
- ***Entomologists*** can learn that until 1921 it was thought that the locust *Schistocerca gregaria* in the ordered and unordered phase were two different species (chapter 6.1.3).
- ***Traders*** can learn how to trade a red paperclip for a two-story house (chapter 6.2).
- ***Game theorists*** can learn how the fairy tail "Hans in Luck" is connected to incremental decision making (chapter 6.2.1).
- ***Revolution leaders*** can learn that Tocqueville thinks that all revolutions are led by madmen (chapter 6.3).
- ***Historians*** can learn how the dynamics of the Monday demonstrations in the former German Democratic Republic may be explained (chapter 6.4).

If you are interested: please read on.

1.4 Relationship to existing FOSS research

There is a wide variety of research about FOSS going on (Cusumano 2005: xi-xii):

- Economists want to know why people and companies are *motivated* to devote scare resources to improve a public good.

- Social scientists are fascinated by FOSS *coordination* mechanisms and development processes.
- Management specialists are interested in how for-profit firms may *take advantage* of the FOSS phenomenon.

The following dissertation includes a mix of these objectives. There is a *motivational* part (see chapter 3 for a review of the corresponding literature) and *coordination* structures and mechanisms are also discussed (see chapter 4 for a review of the corresponding literature). Briefly mentioned are advices for for-profit firms to conserve the goodwill of the community and to *take advantage* of the phenomenon (see chapter 2.6 and 3.4.1 for a review of the corresponding literature). Completely new in this dissertation is the connection of FOSS to the dynamics of collective action.

1.5 Procedure

Chapter ***two*** introduces into the ***FOSS*** phenomenon. It provides a brief account of FOSS development as well as several definitions. A short history of free revealing and sharing delivers insight into the advancement of FOSS. Thereby major examples of FOSS projects such as the Linux kernel or the Apache HTTP server are presented additionally. Afterwards, the complicated but important topic of licenses is discussed in length. To supplement the claim about the importance of licenses the Netscape - Mozilla - Firefox example is discussed. It is also concluded that FOSS is a public good and that path dependencies, switching costs, and network effects play an important role in software as well as in FOSS development.

In chapter ***three*** the problem of ***motivation*** is addressed. To do so self determination theory as well as intrinsic and extrinsic motivation are discussed. Different types of motivations to contribute to FOSS development are pinpointed. Programming just for fun followed by pro-social motivation, peer recognition, human capital, reputation, personal requirement, and earning a salary are discussed. Then an overview over empirical studies on the motivation to contribute to FOSS is given. The chapter is finally complemented by reflections on difficulties between commercial firms and the FOSS community and between paid and non-paid contributors. This is exemplified by the Debian-Dunc-Tank example.

In the succeeding chapter ***four*** the ***organization*** of FOSS development and coordination problems will be discussed. Reflections on governance structures and mechanisms introduce the chapter. Subsequently, important characteristics of FOSS development such as peer review and egoless programming, recruitment of project members, computer-mediated interaction, release early and often, user involvement, modularization, decision making structure and leadership, code reuse, and coordination mechanisms are analyzed. In the following, it is pointed out how FOSS characteristics are observable beyond software development. An excursion shows how structures and mechanisms found in FOSS development played a crucial role in a very different area. The power of self governance and fairness is shown in a comparison between pirate democracy and merchant ship tyranny in the early modern period. This example is also connected to meritocracy, leadership, and decision making structure.

Chapter ***five*** discusses ***collective action*** and social dilemmas in length. After an introduction into wording definitions and classifications are provided. Very basic for the progress of this dissertation is the question how collective action can be nurtured. The disquisition of the factors influencing collective action is structured in individual factors, perceptual factors, decision structure, and social structure. It is argued that the decision sequence respectively incremental decision making is a very prominent factor that is resumed later. Based on these factors different solutions for social dilemmas are compared. There are structural and motivational solutions that are discussed in turn and afterwards contrasted. In an excurse it is investigated whether moral behavior is just another case of selfishness. Subsequently, the production function of collective goods is reviewed. Finally, a connection between social dilemmas and real life business is made.

Chapter ***six*** deals with the ***dynamics of collective action***. Similar concepts such as critical mass, tipping points, thresholds, Schelling's reflection on binary collective decision making, and increasing return technologies are discussed. With PledgeBank a real world example of how critical mass influences decision making is presented. In the following, the influence of incremental decision making is analyzed. With the fairy tale Hans in luck it is shown how value may incrementally degenerate. Similar principles are discussed by pinpointing herding effects, bandwagon effects, and informational cascades. It is then shown that bandwagon respectively similar effects also exist in FOSS development. The important finding that mixed group motivation and group heterogeneity may be beneficial for solving the problem of collective action is then presented. The importance of critical mass and bandwagon effects for successful collective action is exemplified by an excursion that analyses the Monday demonstrations in Leipzig (Germany) in 1989.

Chapter ***seven*** shows that incentives are not only distinct in different production phases of a common good but also in ***organizational life cycles*** and ***in group phases***. It is for example shown that the first phase in organizational life is dominated by creativity and produces other incentives than later phases.

Based on the preceding reflections ***hypothesis*** are generated in chapter ***eight***. It is argued that the relationship between development phases and motivation is *firstly* modulated by network externalities and switching costs respectively by the S-shaped production function of software. *Secondly*, the relationship is heavily affected by incremental or sequential decision making that changes payoffs, improves the information for the successor, allows antecessors to make a credible commitment and fosters conditional cooperation.

In chapter ***nine*** the ***statistical basis*** of the following empirical analysis is discussed. Sample size, significance criterion, effect size, and statistical power are determined. The same holds true for the used statistical tests, methods, survey forms, and sampling procedures.

Chapter ***ten*** presents the ***survey*** itself. To do so the surveyed clusters, the survey items, the measurement model, and data handling are discussed.

The ***results*** of the survey are presented in chapter ***eleven***. In a first step the problem of missing values is properly discussed. In a next step descriptive statistics are used to describe the basic data of the study. Afterwards inferential statistics are used for hypothesis testing. The overall hypothesis that development phases and different types of motivation are related to each other is supported. However, it is found that, surprisingly and contrary to the hypotheses, autonomy as well as fun seeking remain almost constant during FOSS development. Pro social motivation and investment in reputation (contrary to the hypothesis) increase more or less linearly during FOSS development. As hypothesized the motivation related to human capital increases in the initial phase, decreases in the intermediate phase, and increases in the mature phase of FOSS development. Not in line with the hypothesis personal requirement answers to human capital. The relationship between peer recognition and FOSS development is, as suggested, u-shaped.

Chapter ***twelve sums up***, ***discusses the results***, and provides ***suggestions for further research***. It is argued that the found relationship between development phases and developer motivation urges to deliberately adjust governance structures and mechanisms to different types of motivation and development phases. However, based on a comparison between different types of collective action (Monday demonstrations in Leipzig and

FOSS development) it is argued that there is no corresponding silver bullet. In respect to governance structures and mechanisms all arrangements have to be adjusted to the specific collective action in question.

That autonomy and fun seekers remain relatively constant is surprising. This result may be explained by the self-selection mechanism found in FOSS projects: Developers choose themselves to which project they contribute to. If they appraise being constricted in their autonomy fun seekers quite contributing. That is why only fun seekers who do not appraise being constricted remain in a project. In consequence, it is suggested that due to this self-selection mechanism autonomy as well as fun seeking remain constant. This explanation has far reaching implications for commercial projects. It is difficult for commercial firms to reproduce self-selection mechanisms. As a solution to this obstacle commercial firms that intend taking advantage of the FOSS production model should either deliberately foster other motivation types than fun seeking or treat the autonomy of their employees with special care.

It is also concluded that commercial firms that intend to make use of the FOSS production model will hardly get around putting their own source code under a FOSS license. Otherwise, commercials firms cannot take full advantage of user involvement and source code reuse. If source code is released under a FOSS license there is a strong rational to use the GNU GPL or at least a GNU GPL compatible license.

Approach

The economic approach used by Becker (1997) and others does not assume that individuals are motivated only by selfishness. In fact, it is a method of analysis and not an assumption about motivation. It presumes that individuals maximize welfare but does not determine what welfare is. The action related to the maximization of welfare may be called "selfish, altruistic, loyal, spiteful, or masochistic" (Becker 1997: 38). In the economic approach behavior is forward-looking and consistent over time. Individuals struggle to foresee the uncertain consequences of their actions that may include looking back because the past influences attitudes and values for a long time. Economic analysis is therefore related to rationality. Downs (1957: 5) defines rationality:

> *In such analysis {economic analysis}, the term rational is never applied to an agent's ends, but only to his means. This follows from the definition of rational as efficient, i.e., maximizing output for a given input, or minimizing input for a given output. Thus, whenever economists refer to a "rational man" they are not designating a man whose thought processes consist exclusively of logical propositions, or a man without prejudices, or a man whose emotions are inoperative. … Assume that a monk has consciously selected as his goal the achievement of a state of mystical contemplation of God. In order to attain his goal, he must purge his mind of all logical thoughts and conscious goal-seeking. Economically speaking, this purging is quite rational, even though it would be considered irrational … by any of the noneconomic definitions of rationality.*

In contrast to the approach presented above it is in economical writings sometimes at least implicitly assumed that individuals maximize their welfare by being opportunistic. Williamson (1975: 26-30; 1985: 47) defines opportunism in his transaction cost theory in the following way:[17]

> *By opportunism I mean-self-interest seeking with guile. This includes but is scarcely limited to more blatant forms, such as lying, stealing, and cheating. Opportunism more often involves subtle forms of deceit.*

[17] Williamson (1985: 47) also discusses two other behavioral assumptions that are "simple self-interest seeking" and "obedience". However, opportunism is the behavioral assumption that has attracted the most attention.

Clearly, some individuals buy more than they can pay for, embezzle funds, and rob banks (see Williamson 1975: 7).[18] Therefore, the assumption that individuals *may* steal, lye, and cheat is certainly lifelike and realistic.[19] Although the *possibility* of stealing, lying, and cheating does not mean that individuals always show such behavior this is frequently *implicitly* assumed in economical writing.

The behavioral assumption and the applied approach in this dissertation are that individuals are rational and that they maximize their welfare. Related behavior may be selfish, altruistic, loyal, spiteful, or masochistic. It is assumed that opportunism is a possible but not an inescapable manner of behavior. In fact, opportunism is probably less important in FOSS development than in other areas.

The use of examples

In this dissertation there are many smaller and bigger examples related to the discussed topics. Doing so is highly influenced by Allison's (1999) influential book "Essence of decision". This book on the Cuban Missile Crisis is based on three theoretical chapters and three case studies.[20] Nobel Prize winner in physics Richard Feynman is another inspiration for using examples. He reports that he follows equations not mathematically but by an example (1997: 244-245). Feynman argues that thereby it is possible to find errors very quickly by instinct and experience.

The examples used in this dissertation are sometimes related to software development (e.g. Mambo-Joomla!, Netscape-Mozilla-Firefox) and sometimes they are not (e.g. Monday demonstrations, pirate democracy, Hans im Glück). There are different reasons for including examples not related to software development:

i. Such examples show primarily that a discussed phenomenon is not restricted to software development. For instance the excurse discussing pirate democracy and meritocracy shows that some governance structures and mechanisms found in FOSS projects can probably be adopted for other areas beyond software development.

ii. Including examples simplifies the understanding of a phenomenon by connecting it to a set of known knowledge. This may be especially important for readers not familiar with software development.

---

[18] In an experiment money was introduced to a community of tufted capuchin monkeys (Chen & Hauser 2005; Chen, Lakshminarayanan, & Santos 2006; Hauser et al. 2003). The monkeys seem to have understood the concept of money quite well. They were very sensitive to changes in prices, budgets, and expected payoffs (Chen, Lakshminarayanan, & Santos 2006: 534). To a lesser degree they displayed both reference dependence and loss-aversion. They also developed behavior such as stealing, deceit, and prostitution (Dubner & Levitt 2005: 31). In consequence, it may be argued that opportunistic behavior is not restrained to human beings.

[19] In contrast to this line of reasoning some authors note that scientific assumptions do not have to be realistic. Friedman (1966: 14) for example argues that "truly important and significant hypothesis will be found to have 'assumptions' that are wildly inaccurate descriptive representations of reality, and, in general, the more significant the theory, the more unrealistic the assumptions". He further states that a hypothesis must be descriptively false in its assumptions. This is because an important hypothesis explains much by little that makes abstraction necessary. A theory should then not be judged whether it is realistic but whether it is a sufficiently good approximation for the purpose in hand respectively whether its predictions are sufficiently accurate (15).

[20] Chapter one deals with a rational actor model and corresponding hypothesis are developed. Based on the real events these hypotheses are tested in the subsequent chapter. In chapter three an organizational process model is discussed and in chapter four verified. In chapter five a governmental politics model is developed to be tested in the next chapter. This structure pursues different goals. *Firstly*, different theoretical theories are summarized. *Secondly*, the Cuban Missile Crises is explained. *Thirdly*, the strengths and weaknesses of the different theoretical approaches are convincingly shown by testing them in reality.

iii. The examples have also the purpose of making the writing and the reading of this dissertation interesting and agreeable.

1.6 Reservations

Before FOSS is explained in chapter two the following reservations about this dissertation are made.

Community projects vs. projects driven by commercial firms

There are two different categories of FOSS projects: Community driven projects and projects dominated by commercial firms (see chapter 2.3 and 2.3.1). In community driven projects such as the Apache HTTP Server project the software is developed by a public community and the copyrights are owned by individual developers or a non-profit corporation. Although commercial firms may be heavily involved in these projects they do not dominate the road map. In projects driven by commercial firms such as MySQL software is mainly developed by salaried programmers. Contributions from the outside are hardly accepted so that the firm is the single copyright holder. Thereby these firms can put the software under a dual license scheme with a proprietary second license.[21] These projects are almost completely dominated by commercial firms.

The characteristics of these two types of FOSS projects seem to be quite different. It could be assumed that for example the composition of motivation, the recruitment processes, and the development processes of community and company driven projects are diverse. Although produced under FOSS licenses the production model of company projects is perhaps very similar to traditional software development. Because the goal of this study is to research the FOSS development model the following comments *only apply to community driven projects*. The hypotheses and discussion in this study therefore refer solely to community driven projects.

Is FOSS better?

Having chosen FOSS as a topic for this dissertation does not include an implicit argument that FOSS is superior or the like. Research on FOSS sometimes assume (technical) superiority somewhat overly confident and without proper reasoning. It seems to be clear that there is no magic mechanism that makes source code produced by FOSS projects automatically better (or worse) than source code produced by commercial projects. There are certainly examples of very good and very bad FOSS applications. However, a comprehensive comparison between FOSS and commercial software concerning code quality is left to other researchers.[22]

Standing of the rationales

Real world events are always a combination of many different processes as well as accidental occurrences (Marwell & Oliver 1993: 14). Probably accidental occurrences are more important explaining success or failure of a FOSS project than anything else. Despite the importance of fortune and chance they are rarely addressed.

[21] Legally such a license scheme with a proprietary second license is also possible if copyrights are broadly distributed. However, because every single copyright holder has to accept the proprietary license it is practically not possible. For a discussion of dual licensing see Välimäki (2005: 206-216).

[22] Benkler (2002: 403) argues that the peer production model will prevail if its costs are lower than using the market or hierarchical organization.

The reason seems to be quite clear. Fortune and chance cannot be influenced and are therefore of minor scientific as well as practical interest. Similarly, the influential factors fortune and chance are not addressed in this dissertation although they are thought to be very important.

As we know from studies on organizational performance it is very difficult to include all relevant variables explaining success or failure (March & Sutton 1997). FOSS development is a dynamic and complex phenomenon. It seems to be therefore obvious that the observation made for studies on organizational performance is also true for FOSS research. Additionally, success factors of FOSS projects may change from one project to another (Robles 2004: 196). In consequence, there will be never a completely reliable cooking recipe that explains the success of FOSS development. In this study only selected aspects of the FOSS phenomenon are illuminated. It is not assumed that they are the *most* important for FOSS development. However, it is thought that *they are* important.

Usefulness of collective action

In the following chapters it will be discussed how groups manage collective action. Doing so does not imply that all collective goods are beneficial for a larger group or society as a whole. The reputation of being tough to the ones who do not pay protection money is a public good for the members of the Mafia. At the same time a society as well as ordinary citizens may be hindered enormously in their economic advancement and in their personal integrity if the Mafia succeeds to produce the mentioned public good (see also Gambetta 1988). Here it is in the public interest that cooperation between the actors collapses. For the argumentation here only the technical question how a collective good can be produced is relevant. The question whether a collective good is beneficial for society or not is not answered. It should not be taken for granted that this is always the case.

Reasoning

In this dissertation it is attempted to argue based on facts and rational thoughts. Thereby the line of reasoning is thought to be comprehensible and primarily replicable. However, in the end there are many assumptions that could have been made otherwise. One should therefore be careful to accept rational thoughts presented in this dissertation uncritically.[23] At the same time one should recognize that the basic thought presented in this dissertation is more valuable than the single hypotheses.

---

[23] There are two interesting portrayal that show that one should not have too much confidence in rationality. The *first* is a statement by Robert McNamara who served as United States Secretary of Defense and is considered as “one of the 20th century’s best known advocates of rationality in human affairs” (Blight & Lang 2005: 84). McNamara stated in reference to the Cuban Missile Crisis in the documentary ‘*The fog of war*’: “I want to say, and this is very important: at the end we lucked out! It was luck that prevented nuclear war. We came that close to nuclear war at the end {gestures by binging thumb and forefinger together until they almost touch}. Rational individuals, Kennedy was rational, Khrushchev was rational, Castro was rational. Rational individuals came that close to the total destruction of their societies” (Blight & Lang 2005: 59). The *second* portrayal is reported by Nobel laureate in Physics and member of the Rogers Commission on the Space Shuttle Challenger accident Richard Feynman. He acquaints that the management of the NASA estimated the probability of a failure of the Space Shuttle with loss of vehicle and of human life 1 in 100,000 (Feynman 1986: F1). Reality tragically showed with two disasters within 113 shuttle flights (Seife 2003: 1001) that the “management's fantastic faith in the machinery” (Feynman 1986: F1) was very wrong.

The fundamentals of the argumentation

In the following chapters two to five the basics for the proceeding argumentation are developed. ***Chapter two*** introduces the object of this study, namely the FOSS phenomenon. In ***chapter three*** motivation as a primary force for individual behavior is discussed. Governance structures and mechanisms promote or restrict individual behavior and are therefore crucial for explaining a social phenomenon. In consequence, ***chapter four*** deals with governance structures and mechanisms. Special attention is given to governance structures and mechanisms in FOSS projects. Social dilemmas and the problem of collective action are very basic challenges of human interaction. Therefore ***chapter five*** presents the structure of the dilemma as well as possible solutions to the corresponding problems.

2 FOSS: The object of the study

A study analyzed the use of FOSS in the U.S. Department of Defense (DOD) and what could happen if it would be banned (MITRE 2003). The study concludes that within the DOD FOSS plays a critical role in network security, infrastructure support, software development, and in research (MITRE 2003: 2):

- ***Network security:*** The study concludes that network security rests highly on FOSS (e.g. OpenBSD, SARA, and Snort) and that its banning would have immediate, broad, and strongly negative impacts on the ability to defend against cyberattacks (MITRE 2003: 2). The report states further that commercial equivalents to FOSS web products are relatively immature and increase risks for the DOD infrastructure (MITRE 2003: 20). One aspect why FOSS is influential for network security is the ability to change source code in response to attacks (MITRE 2003: 18).
- ***Infrastructure support:*** The removal of FOSS applications (e.g. Apache, BIND, Samba, and Sendmail) would result in a strong negative impact on the ability to support web and Internet based applications (MITRE 2003: 3). The study states that FOSS is generally "functionally mature, and less likely to fail than much more recent proprietary equivalents" (MITRE 2003: 17).
- ***Software development:*** The report concludes that "software development would be hit hard" by a ban of FOSS (e.g. GCC, GNAT, CVS, and Perl MITRE 2003: 3 and 21). Rigorously enforced such a ban could bring affected software development projects to a halt. Especially the programming language Perl would be difficult to replace.
- ***Research:*** The ban of FOSS (e.g. Colt, EADSIM, gnuplot, and R) would also hinder research activities of the DOD. The support costs for research would increase and the ability to share research results in the form of executable software would be lost (MITRE 2003: 3). FOSS provides important "resources such as mathematical software and the ability to link PCs into supercomputers for which there are no equivalent commercial alternatives" (MITRE 2003: 18). The study states that "FOSS software acts as a sort of 'active publication' medium" (MITRE 2003: 20). The negative impacts would be especially grave in numeric processing and simulation as well as in the area of super computer networks with low-cost PC's (MITRE 2003: 20).

Based on this study it can be concluded that FOSS achieved a mature stage making it worthwhile to study the phenomenon.

Behlendorf (1999: 158-160) argues that FOSS tends to be geared towards the infrastructural and back-end side[24] and less towards the front-end side of the software spectrum.[25] At the same time FOSS products enjoyed

[24] Examples of successful FOSS projects at the infrastructural and back-end side are the ones that belong to the so-called LAMP software bundle. Michael Kunze (1998) coined the acronym LAMP in an article on web publishing connected to databases with the title "Let there be light". **LAMP** stands originally for **L**inux, **A**pache, **M**ySQL, and **P**HP. With the evolving of the term Perl or Python were used as substitutes for PHP. Linux is the operating system, Apache the HTTP server, MySQL the database management system and PHP, Perl, or Python the scripting respectively programming languages of the LAMP bundle.

a higher adoption rate in the corporate computing market than in the personal computing market (Feller & Fitzgerald 2002: 113). Although there are counterexamples such as GIMP[26] there is much more sound software related to operating systems and network services than sound desktop applications. For Behlendorf (1999: 158-159) this is due to the following matters of fact:

- During its history FOSS has mainly evolved in the networking code and operating system space.
- FOSS tends to flourish where incremental change is rewarded what applies more to back-end than to front-end software.
- End-user applications are hard to write because one has to engage with a constantly changing nonstandard graphical and windowed environment that has a high complexity. Additionally, most programmers are not good graphical interface designers.
- Many examples of FOSS have been written by software engineers to solve a problem they encounter themselves that is likely not related to front-end software with a convenient GUI.

There are sometimes overly enthusiastic claims that FOSS is per se better. Users of FOSS solutions have probably experienced that there are also bad FOSS applications. However, the opposite argument that software developed by commercial projects is always better is also proved wrong by various good FOSS projects and very bad commercial software. It seems to be therefore quite clear that there is good and bad FOSS as well as good and bad commercial software.

It is also sometimes argued that FOSS imitates proprietary software without innovating new solutions. *Firstly*, there are innovative FOSS projects such as for example Perl, Python, or TeX (see Stallman 2004b). *Secondly*, sequential innovation has not been introduced to the software industry by FOSS (Bessen & Maskin 2002). The spreadsheet solution Microsoft Excel is built on Lotus 1-2-3 that is in turn built on VisiCalc. *Thirdly*, even if the argument were true: so what. On the one hand imitation is per se economically not always bad. On the other hand a firm imitating the success of other firms may become an innovator itself. Long ago Japanese car manufacturer were said to imitate only western car manufacturers. However, since more than 20 year U.S. car manufacturers struggle to imitate Toyota (Hamel 2006: 74; Spear & Bowen 1999: 97).

2.1 Description and definitions

Describing and defining the FOSS phenomenon is not an easy task. It is foremost a specific mode of software distribution. In consequence, FOSS could be solely described by legal terms. However, doing so is too constricted because FOSS is also a software engineering, psychological, philosophical, social, cultural, political, economic, and a managerial phenomenon (Feller et al. 2005: xviii). Therefore, the reader should supplement the following description more geared towards a legal understanding of FOSS with the comments in chapter three and four.

---

[25] In software development the front-end refers to the software part that deals with users while the back-end processes the inputs form the front-end.

[26] GIMP is a bitmap graphic editor similar to Adobe Photoshop.

What is FOSS?

At the very heart of FOSS movement is the principle that the source code of the software is available so that it is possible to understand and modify it (Feller & Fitzgerald 2002: 12). Software is written as a set of instructions written in a programming language such as Ada[27], C, C++, Haskell, Java, or Prolog. These instructions are the source code that is readable by human beings. However, to run software the source code must be converted respectively compiled into machine code. Such machine code is most basically in a binary form and consists of strings of 1s and 0s. A computer can at the most fundamental level only deal with electrical switches which are represented by 1 and 0 (Feller & Fitzgerald 2002: 9): While computers may manipulate such binary configurations many millions of times per second they can hardly be interpreted by human beings. It is very hard to decipher even small programs in binary form (Feller & Fitzgerald 2002: 11).

Source code in C ⇒	**Machine code (in hexadecimal)** ⇒	**Screen output**
#include <stdio.h>	5589e583ec18895dfc83	012
int main()	e4f031db895c240443c7	
{	042434840408e809ffffff	
int counter = 0;	83fb027eea8b5dfc89ec	
while (counter < 3) {	31c05dc3	
printf(„%d", counter);	etc.[28]	
counter = counter + 1;		
}		
return 0;		
}		

Figure 1: A computer program counts from zero to two[29]

The importance of having access to the source code and not only to the machine code has come to wider audience by the discussion around the Joint Strike Fighter primarily developed and funded by the U.S. and the U.K. The U.S. refused to give the U.K. access to the source code of the software that controls the jet. However, as Under Secretary of State and Minister for Defence Procurement Lord Drayson argues the jets are "useless without control of the software as they could effectively be 'switched off' by the Americans without warning" (Chapman 2006). In fact, without having access to the source code one cannot be sure whether the software producer has built in functionalities or bad written source code not appreciated by the user.

There are two different definitions of FOSS: the open source definition of the Open Source Initiative (OSI) and the free software definition by the Free Software Foundation (FSF). Both definitions will be discussed in turn.

[27] See also footnote 46.

[28] These 17 machine instructions are only a tiny fraction of the whole machine code that executes the commands of the source code. The complete machine code for executing the source code to count from 0 to 2 does have over 90'000 machine instructions.

[29] Luethi (2004: 90-91).

Open source definition

According to the OSI (2006b) open source software licenses must comply with the subsequent criteria:

i. The software must be freely distributable.

ii. The software must include either the source code or a well publicized offer to provide it for free or for its reproduction costs.

iii. The license must allow modifying the software and distributing it under the same terms as the license of the original software.

iv. Licenses may require that modifications are redistributed only as patches and that derivative work carry a different name or version number from the original software.

v. There must be no discrimination against any person or group of persons.

vi. There must be no discrimination against any filed of endeavor.

vii. The rights attached to the software must apply to everybody whom it is redistributed without the need for execution of an additional license.

viii. The rights attached to the software must no be specific to a product.

ix. The license must no place restrictions on other software that is distributed along with the licensed software.

x. The license must be technology neutral and not predicate any individual technology or style of interface.

Free software definition

According to the FSF (2006a) software qualifies as free software if its license terms ensure to run, copy, distribute, study, change, and improve it. This refers to the following four kinds of freedom:

- Freedom 0: The freedom to run the program, for any purpose.
- Freedom 1: The freedom to study how the program works, and adapt it to the own needs. The precondition for this freedom is access to the source code.
- Freedom 2: The freedom to redistribute copies.
- Freedom 3: The freedom to improve the program, and release the improvements to the public.

Richard Stallman (2004a) states at the 20th anniversary of the free software community:

> *My hope was that a free operating system would open a path to escape forever from the system of subjugation which is proprietary software. I had experienced the ugliness of the way of life that non-free software imposes on its users, and I was determined to escape and give others a way to escape. … Non-free software carries with it an antisocial system that prohibits cooperation and community.*

Free software versus open source software

The difference between the free software and the open source software definition will be discussed in length in chapter 2.3.2. In a nutshell: the distinction between the two definitions is the "way of looking at the world" (FSF 2006g) respectively tactics (Raymond 1999b). The free software community is based on an ethical position that demands freedom for software while the open source community is based on the production model (Feller & Fitzgerald 2002: 41). Despite this sharp distinction there is almost no difference in what qualifies as free software or open source software. Most licenses accepted as a free software license are also accepted as an open source software license and vice versa.

What is a hacker?

In the following discussion sometimes the term ***hacker*** is used. A very common mistake made in public is to consider that a hacker is a software programmer who breaks illegally into computer systems.[30] However, a programmer breaking into computer systems is called a ***cracker***. Within the programmer community a hacker is not a negative but a very positive term. According to the Jargon File (Raymond 2003, hacker) it refers to the following:

1. *A person who enjoys exploring the details of programmable systems and how to stretch their capabilities, as opposed to most users, who prefer to learn only the minimum necessary. RFC 1392*[31]*, the Internet Users' Glossary, usefully amplifies this as: A person who delights in having an intimate understanding of the internal workings of a system, computers and computer networks in particular.*
2. *One who programs enthusiastically (even obsessively) or who enjoys programming rather than just theorizing about programming.*
3. *A person capable of appreciating hack value.*[32]
4. *A person who is good at programming quickly.*
5. *An expert at a particular program, or one who frequently does work using it or on it; as in 'a Unix hacker' (Definitions 1 through 5 are correlated, and people who fit them congregate).*
6. *An expert or enthusiast of any kind. One might be an astronomy hacker, for example.*
7. *One who enjoys the intellectual challenge of creatively overcoming or circumventing limitations.*

According to Himanen (2001: ix-x) the hacker ethic is based on three characteristics. *Firstly*, hackers have a passionate and enthusiastic relationship to their work. *Secondly*, the motivation to contribute to software

[30] For example the known book "*The cathedral and the bazaar*" of Raymond (1999a) about software engineering in FOSS projects is indexed at the Library of the University of St. Gallen (Switzerland) under the subject "computer crime" (IDS St. Gallen 2007).

[31] RFC (Requests for Comments) are a series of documents encompassing research, innovation, and methodologies applicable to the internet. The RFC published by the Internet Engineering Task Force (IETF) are considered the standard for the internet. RFC 1392 (Network Working Group & Malkin 1993) has been replaced by RFC 1983 (Network Working Group & Malkin 1996). However, the definition of a hacker remains identical.

[32] Hack value refers to something worth doing or interesting although it may be seemingly useless (Raymond 2003, hack value). An example may be the ability of the programming language Maclisp to read and print Roman numerals that was only implemented for hack value. It also refers to a solution that is challenging for a programmer and is thought to be impossible.

projects is to create something that is valuable in one's own peer community. *Thirdly*, there should be easy access to information and computer resources.

Hackers probably suffer from the "computer disease" as Nobel Prize winner in physics Richard Feynman (1997: 127) positively puts it.[33] To pinpoint the disease he narrates the real example of somebody who tried to calculate arc-tangent automatically with the help of a computer and make a table out of it in one operation. While this task was not trivial it was "absolutely useless" because there were already arc-tangents tables. The disease is then based on "the delight in being able to see how much you can do".

For Levy (1984: 40) the very root of hacker ethics is that "access to computers – and anything which might teach you something about the way the world works – should be unlimited and total". Hacker ethic includes also mistrust of authority and the promotion of decentralization (Levy 1984: 41-43). Hackers should be judged by their hacking and not by degrees, age, race, or position (Levy 1984: 43). In hacker communities it is assumed that one can create art and beauty on a computer and that computers can change one's life for the better (Levy 1984: 43-49).[34]

While hackers are mostly associated with software development the term may also be associated to any activity one pursues with passion. The designer of the Macintosh computer Burrell Smith noted that "hackers can do almost anything and be a hacker. You can be a hacker carpenter. It's not necessarily high tech. I think it has to do with craftsmanship and caring about what you're doing" (reported in Levy 1984: 434).

2.2 A short history of free revealing and free sharing

Free revealing of source code was widely common in the early days of the computer (Feller & Fitzgerald 2002: 28). It was common to share source code in IBM or DEC user groups (Feller & Fitzgerald 2002: 28; Hars & Ou 2002: 26). Such free revealing was accompanied by intensive peer-reviewed development that was deeply rooted in the Unix culture from its beginning (Raymond 2004: 7).

A reason why source code was usually revealed is perhaps that computers were first used as research tools at universities (see Perens 1999b: 172). Raymond (2004: 45) notes that for the early ARPANET hackers sharing source code was something close to a religion arising form the academic "publish or perish" imperative. The motivation for free revealing was sometimes also based on efficiency considerations. An example is the PACT (Project for Advancement of Coding Techniques) initiative in which competitors shared source code. The software produced by the user consortium PACT was considerably better than anything IBM offered (Campbell-Kelly 2003: 33).

In the following important projects and milestones related to free revealing and free sharing are discussed in chronological order.

BSD

A single event that boosted free revealing and free sharing was a U.S. government decision that outlawed the telecommunication firm AT&T from entering into non-telephony markets. As a consequence, AT&T could not

---

[33] The 3.0.x releases of Scientific Linux Fermi LTS, a Linux distribution based on Red Hat Enterprise Linux, are named after Richard Feynman (Dawson 2008).

[34] For a short discussion of the importance of beauty in programming see footnote 230.

sell its Unix operating system and gave it away for free to universities and research institutions. The Unix source code was then subsequently advanced at the University of California at Berkeley. Starting in early 1977 (McKusick 1999: 33) the Berkeley Software Distribution (BSD) got one of the most important distribution channels for Unix and related software (Feller & Fitzgerald 2002: 29). The BSD group developed many important parts that still run the Internet today. Examples are the TCP/IP protocol suite as well as BIND which is the “most mission-critical software underpinning the whole internet” (Feller & Fitzgerald 2002: 30). BIND was written by four students at the University of California at Berkeley. It maps domain names (e.g. http://www.google.com) that are understandable by human beings to IP addresses (e.g. 66.102.9.104) that are understandable by computers.

TeX

A main figure in FOSS history is Donald Knuth who is “one of the true giants in the history of the computing field” (Feller & Fitzgerald 2002: 30-31). He contributed the typesetting software TeX, that is widely used in mathematics and physics, to the public domain.[35] Similar to the peer review system in the FOSS production model Knuth asks to report bugs in his books and offers to pay a hexadecimal \$ (2^8 or 256 cents) respectively 2.56 \$ for each bug. Due to this bug reports Knuth’s books are considered *almost* free of error.[36]

FSF

The establishment of the Free Software Foundation (FSF) by Richard Stallman was also very important in the FOSS history (Feller & Fitzgerald 2002: 32). The FSF was rooted in the sharing of source code at MIT’s Artificial Intelligence Laboratory but also elsewhere (Stallman 2001a). However, in the early eighties the software sharing community collapsed. Because this community was gone Stallman quitted his job and decided to start the GNU project and to write a free operating system. With the time other people joined and the FSF was founded to get a tax - exemption.

Linux

The success of the Linux operating system[37/38] was certainly a major milestone in FOSS history. Linus Torvalds announced Linux as a “(free) operating system (just a hobby, won't be big and professional like gnu) for

---

[35] More than 90% of books published in mathematics and physics use $\TeX$ (Feller & Fitzgerald 2002: 31).

[36] However, the monetary value of the checks does not seem to be the real incentive for the bug reports. Although Knuth (2002: 324) has issued more than 2000 checks with an average value above 8$ very few checks seem to have been cashed. Most checks are framed and put on the internet. Ditlea (2002) argues that “Knuth's reward checks are among computerdom's most prized trophies”.

[37] There is a controversy whether the operating system should be called Linux or GNU/Linux (Raymond 2003, Linux; Stallman 2002). The operating system consists of the Linux kernel, many GNU components such as the GNU C Library, and contributions from other projects such as the X Window System. To complicate the naming further there are also Linux distributions such as Fedora Core, SUSE Linux, Ubuntu, Mandriva Linux, Debian, or Gentoo that are sometimes also simply called Linux. These distributions include not only an operating system but also other software such as compilers, productivity tools, text editors, web-browsers, or scientific tools. In this study the kernel is addressed as Linux kernel. The operating system is addressed, for simplicity and without taking up a stance on the naming controversy, as Linux. The Linux distributions are addressed by their full name such as SUSE Linux.

[38] The original name for Linux was Freax but Ari Lemmke asked Torvalds to use the name Linux (Feller & Fitzgerald 2002: 33).

386(486) AT clones” to “everybody out there using minix” (Torvalds 1991). Since then Linux has become tremendously successful and is probably the most influential FOSS application. According to the market research firm IDC (2006) that is specialized inter alia on information technology “Linux servers posted their fifteenth consecutive quarter of double-digit growth, with year-over-year revenue growth of 17.0% and unit shipments up 14.4%”.[39] In the first quarter 2006 Linux represented 12.2% of the total 11.9 billion $ revenues in the server market. At the same time Unix servers had a share of 33.2% and Windows of 37.1%. While Unix servers experienced a decline of 7.1% Windows grew 5.9%. From this figures it can be inferred that the growth of Linux is at the expense of Unix and not of Windows.

According to the Linux senior strategy manager at IBM there are three main industries with the most traction for Linux (Solheim 2006):

i. ***Finance:*** Linux is suited for Wall Street because of its need for faster computing power and because of the flexibility of technology (e.g. phone or web) to engage with customers. An example is Charles Schwab that has chosen Linux to do investment calculations faster and answer client queries on the phone (Stuckhoff 2005).

ii. ***Retail:*** Another industry for which Linux is apt is retail business because of technological flexibility and customization (e.g. checkout). Pioneer Petroleum is an example that uses Linux on its point-of-sale systems at their gas stations (Stuckhoff 2005).

iii. ***Government:*** Governmental authorities are suited for Linux because of their need to reduce costs and provide new system services. The city of Bergen is an example of a government looking to reduce costs by implementing Linux in public schools (Stuckhoff 2005).

However, Linux is also used in other areas such as by DaimlerChrysler for simulation of car crashes, by Pixar Animation Studios and DreamWorks for digital animation of films (Monsters Inc., Toy Story, and The Incredibles; Shrek 1 to 3, Madagascar), by oil explorer Amerada Hess for number crunching to find oil, or by Covertex to compute the cushion structure[40] of the soccer stadium Allianz Arena in Munich (Ante et al. 2003; Novell 2005: 214; Rowe 2007). At the same time the adoption of Linux on the desktop is in its very early days (Ante et al. 2003; Solheim 2006).

Apache

The development of the Apache HTTP server begun in 1995 by volunteers who decided to pool their expertise and effort (Feller & Fitzgerald 2002: 34-35). The reason for this coordination was that the programmers of the National Center for Supercomputing Applications stopped the developing of their popular web server. The center of development gone the users had to help themselves.

---

[39] In this dissertation very often percentages are reported. This is not without pitfalls. A funny example is reported by the Economist (1998: 70): The Mexico City government increased the capacity of a motorway from four to six lanes by re-painting lines. This resulted in a 50% increase in capacity. However, this caused an increase in serious accidents and after a year the motorway was changed back to a four-lane road. In consequence, the motorway capacity decreased by 33%. To support its claims of progress the government subtracted the 33% reduction in motorway capacity from the 50% increase to announce a net increase in motorway capacity of 17%.

[40] The Allianz Arena facade is made of 2816 individual rhomboid ETFE-foil cushions (Covertex 2003).

Apache plays a key role in the World Wide Web. It runs on different operating systems and is used to serve static and dynamic content.

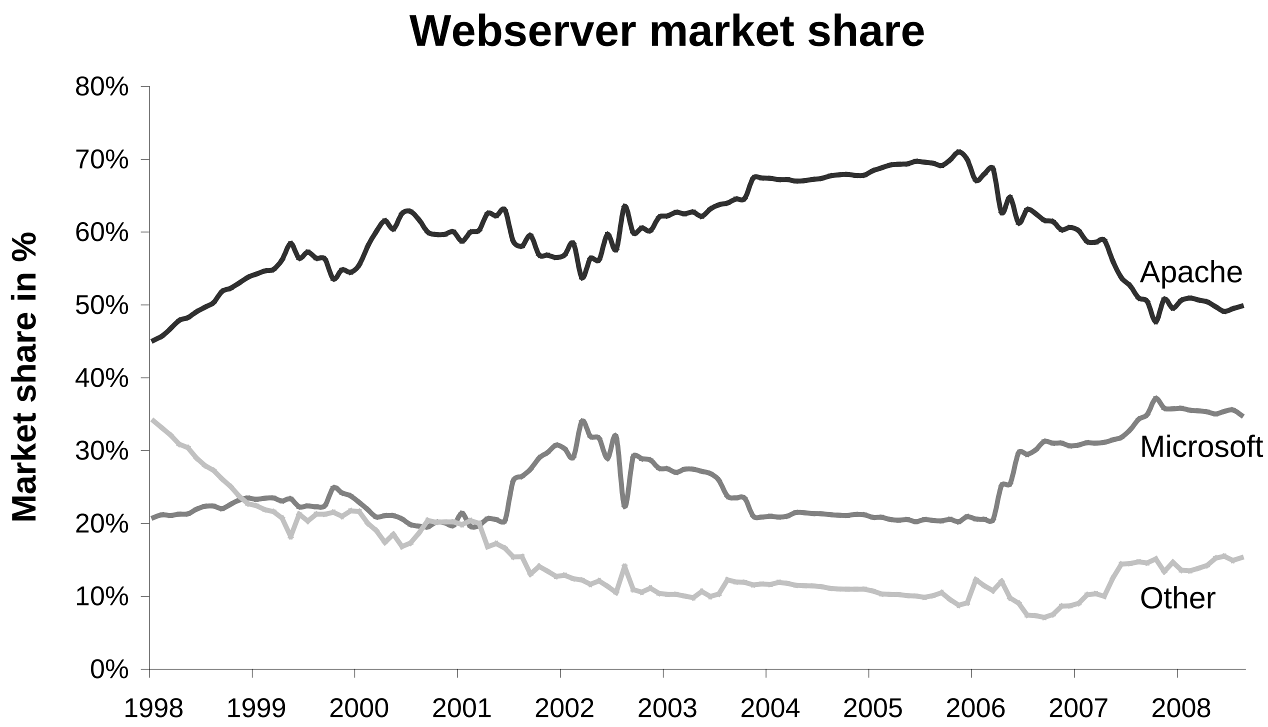


Figure 2: Web server market share[41]

According to Netcraft (see Figure 2) Apache servers hosted in August 2008 50% of all domains while Microsoft's IIS hosted 35%. An advantage of Apache is its massive stability (Feller & Fitzgerald 2002: 22).

The Apache HTTP server is a special FOSS project for several reasons. *Firstly*, it was from the very beginning developed by and for salaried programmers. Commercial firms and individuals who had to run a web server contributed to solve their own needs. *Secondly*, project structure and processes are highly formalized and decision making is based on voting (Apache Software Foundation 2006b).

Netscape-Mozilla-Firefox

What observers commented as "freeing the lizard" [42] was the move of Netscape in 1998 to put its web browser Communicator 5.0 under a FOSS license (Young & Rohm 2000: 64). In this highly noted step Netscape created the FOSS project mozilla.org to develop the application further. The importance of this project for FOSS history was not rooted in the application itself. In fact, for a very long time the success of the project was very questionable (see chapter 2.5.3). The project recognition in the FOSS community was based on its symbolic value. It was the first time a large software company made its source code free (Netscape 1998). Similarly,

[41] Netcraft (2008). The decline of market share after April 2006 is due to migration of several million hostnames of the domain registrar Go Daddy from Apache to Microsoft (Netcraft 2006b). At the same time Apache has overtaken Microsoft as the leading secure web server (Netcraft 2006a).

[42] A Godzilla like lizard is the mascot of the Mozilla project.

O'Reilly (1999: 33-34) observes that the open source campaigning became international news because of Netscape's decision.

Initially, the Mozilla project did not progress. Among other developers Dave Hyatt and Blake Ross were not happy with the Mozilla web browser because they thought that it had an unnecessarily large amount of features. In consequence, they started an experimental fork. They named the new web browser Firefox and got by now very successful. In summer 2006 the global market share of Firefox was approximately 10% (Wikipedia 2007c).[43] There are huge differences in usage. While in Germany Firefox had a market share of 39% in July 2006 the share in the US was 16% (OneStat.com 2006). For the website of Heise online, a site for people interested in IT, Firefox had in June 2006 more than 50% market share (Heise online 2006). Roughly 38% of the readers of Spiegel online, a German weekly magazine, use Firefox during working days, on weekends this share mounts to 46% (Patalong 2006).

OSI

In 1998 the Open Source Initiative (OSI) was founded to give the meaning of FOSS a new direction. The founders of OSI coined the term open source software as a contrast to free software. The rational behind this move was the fear that the free software community and its terms do not have enough appeal for commercial firms. The OSI wanted "to market the free software concept to people who wore ties" (Perens 1999b: 173). The differences between Free software and open source licenses are practically marginal (see also chapter 2.3.2). However, the attitude towards free revealing is completely different.

2.3 FOSS project typology

While there are certainly many important FOSS characteristics (see chapter 4) two are outstanding for this study (see Figure 3):

- *Project control:* Community driven projects versus projects dominated by commercial firms.
- *Weltanschauung:* Free software versus open source software

Projects developing office applications, projects developing development tools, or projects of any other major area of development are certainly different. In consequence, projects are also distinguishable in respect to the type of application that is developed. Projects differ from each other also depending on the involved actors.

[43] Measuring browser market share properly is not trivial because of techniques such as refreshing, feed reading, link prefetching, or ad blocking. Some browsers such as Opera are able to hide their true identity. However, the presented estimations of Firefox market share seem to be conservative (Settele 2006: 61).

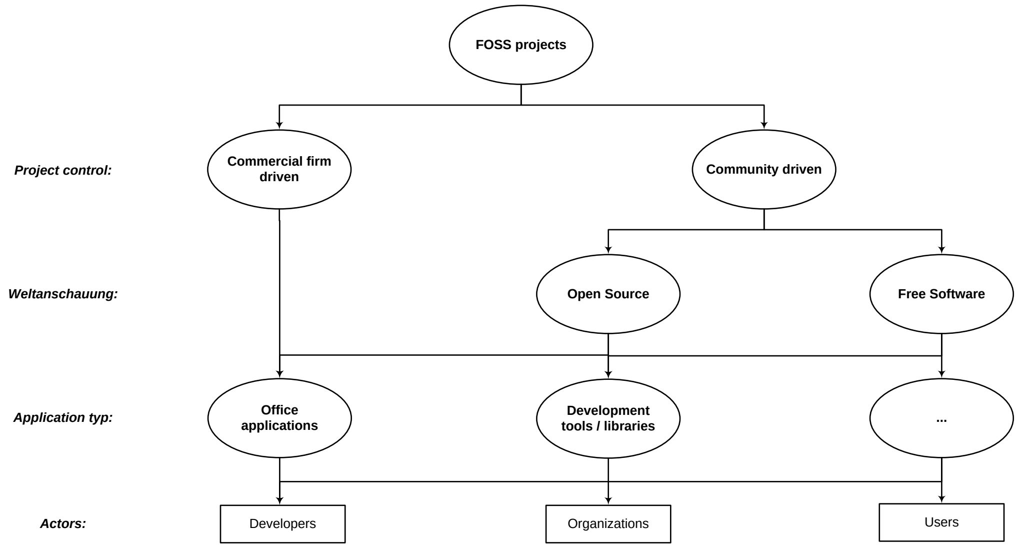


Figure 3: FOSS project typology

In the following, the two main characteristics project control and Weltanschauung are discussed in turn.

2.3.1 Project control: Community versus commercial firm driven

In community driven projects it is a community that primarily develops software and makes decision. According to Raymond (1999a: 62) a viable community is important for the success of a FOSS project: although “coding remains an essentially solitary activity, the really great hacks come from harnessing the attention and brainpower of entire communities.”

Community affiliation is usually neither formal nor are there clear boundaries. Although it is very difficult for an outsider to define community affiliation properly the affected individuals probably know themselves whether they belong to a specific community or not.

In community projects it is not excluded that commercial firms add important contributions like for example IBM for the Linux kernel. These commercial firms, however, dominate neither the road map nor the development processes. Furthermore, they are expected to behave as members of the community. Goldman and Gabriel (2005: 9) argue that {for managers} “one of the hardest lessons to learn about working in open source is that the work involves community building, politics, citizenship, principles, and governance”.

In projects driven by commercial firms the road map, development processes, and user communities are dominated by firms. In some of these projects such as MySQL only user but not developer communities are welcome. Accepting contributions from outsiders is accompanied by a loss of control of copyrights that in turn makes it difficult to put the software under a proprietary license in a dual license scheme. In such cases commercial firms foster a good relationship only to user communities. However, this is also vital for proprietary software not under a FOSS license or any other product. Because of this it is argued that communities are in such project not a central element of the software development process.

2.3.2 Weltanschauung: Free software versus open source software

In community driven FOSS projects there are two main communities: the free software and the open source community (see Figure 3). Before discussing differences and similarities it has to be stated that not for all projects a clear classification can be made. In certain community driven projects some developers consider being member of the one or the other community or do not care at all. Certainly, there are also projects that belong clearly to the free software community (e.g. GNU toolchain) or to the open source community (e.g. Apache or Perl). Despite this reservation and the related ambiguity the distinction between the two different communities seems to be very important.

Both communities consider more or less the same licenses as free respectively open. A license accepted as a free software license does most probably also qualify as an open source license and vice versa (FSF 2006g). Therefore, the lists of free software licenses and open source licenses are almost identical. The only known license that qualified as an open source license but not as a free license was the Apple Public Source License prior to version 2.[44] However, version 2 of this license is considered as well as a free software license (FSF 2006b). For this reason any free software license is an open source license as well and vice versa.

The difference between free software and open source software is its Weltanschauung respectively the "way of looking at the world" (FSF 2006g). The open source community considers access to source code as a practical question while the free software community considers it as an ethical question (FSF 2006g). The free software movement is mainly concerned that software is or remains free. Free is according to this Weltanschauung "a matter of liberty, not price" and the FSF (2006a) urges that one should think of "free as in free speech" and "not as in free beer" which relies to free of charge. The FSF (2006g) calls the open source movement a methodology of developing software while they call the free software community a social movement. A strict member of the free software community would not contribute to a proprietary software project even it had outperform the next best free software solution considerably. An example of such an attitude is free software advocate Richard Stallman who started to build a new operating system. He intended to develop "a kernel plus all the utilities needed to write and run C programs: editor, shell, C compiler, linker, assembler, and a few other things. After this we will add a text formatter, a YACC, an Empire game, a spreadsheet, and hundreds of other things" (Stallman 1983). With other words: Stallman intended to replace every software program without being concerned that there were reliable proprietary equivalents. A member of the open source community would probably contribute to a proprietary project if the development style were similar to the one in FOSS development. While the FSF and the free software community is driven by an ethical position that demands freedom for software, the OSI and the open source community is driven by a pragmatic position that is based on the superior production model (Feller & Fitzgerald 2002: 41).[45]

The founders of the OSI thought that the FSF is not business friendly enough. They intended promoting the less ideologically stamped term open source instead of free software. It was thought that by doing so more commercial firms and users could be attracted. The open source advocate Eric Raymond (1999b) argues that the difference between the two communities is in tactics. While the free software community advocates "rhetoric" about freedom the open source community advocates persuasion by superior software. Raymond concludes that Stallman's "best propaganda has always been his hacking" and instead of talking about freedom one should shut up and "show the code". Interestingly, Bruce Perens, a co-founder of the OSI and author of the open source definition, regretted some effects of the popularity of the open source term (Perens 1999a).

[44] The kernel of the operating system Mac OS X (Darwin) is licensed under the Apple Public Source License (Apple 2006).

[45] Interestingly, Microsoft (2005a) as a sharp opponent of FOSS seems to agree on the basics of this distinction. However, the FSF is probably not happy with the chosen wording and with the further arguments of the cited reference.

According to Perens the open source movement de-emphasized the importance of the freedoms associated with free software, overshadowed the FSF efforts and shifted away from the basic values.

In either case Stallman (2001b) writes that radical groups usually split because of disagreement on details and treat each other as enemies. However, the relationship between the free software and the open source community does, according to Stallman, not fit this observation: The communities disagree on the basic principles but agree on practical recommendations. Stallman writes further that both communities are not enemies and that they can work together.

2.4 Who contributes and what is contributed?

The following subchapter presents demographics about FOSS developers and empirical findings about the contributions. The corresponding data is based on several FOSS surveys discussed in length in chapter 3.3.

Gender

According to different studies there are almost no women who contribute to FOSS projects (David, Waterman, & Arora 2003b: 8; Ghosh et al. 2002a: 8; Hars & Ou 2002: 31; Hertel, Niedner, & Herrmann 2003: 1167; Lakhani & Wolf 2005: 8; Luthiger 2006: 73; Mitsubishi Research Institute 2004c, 2004d; Robles et al. 2001a). Depending on these studies the percentage of female developers is between 1.1% and 5% (see Figure 4).

Study	**Percentage of female FOSSS developers**
FLOSS (Ghosh et al. 2002a: 8)	1.1%
WIDI (Robles et al. 2001a)	1.4%
FLOSS-Asia (Mitsubishi Research Institute 2004c)	1.5%
FLOSS-US (David, Waterman, & Arora 2003b: 8)	1.6%
Luthiger (2006: 73)	1.9%
FLOSS-JP (Mitsubishi Research Institute 2004d)	2.0%
BCG (Lakhani & Wolf 2005: 8)	2.5%
Linux kernel (Hertel, Niedner, & Herrmann 2003: 1167)	4.3%
Hars and Ou (Hars & Ou 2002: 31)	5%

Figure 4: Percentage of female developers in different FOSS studies

The percentage of women in computer science is generally low in most countries (Galpin 2002: 97-98). Typically, the percentage of female programmers is in western societies between 10% and 20%. In some countries such as Guyana, Singapore, Thailand, Iran, and Zimbabwe the fraction of female programmers is higher. Therefore, it can be concluded that the fraction of women in FOSS development is not only considerably lower than in the population but also lower than in the domain of computer science. This is astonishing because in FOSS development the barriers for participation are low.

Interestingly, programming in the very early history of the computer was heavily influenced by women (Goyal 1996). It was Augusta (Ada)[46] Byron, Lady Lovelace, who wrote a description for using Babbage's Analytical Engine (Toole 1996: 4). This description is widely considered the first program and Ada Byron the first programmer. A further remarkable female programmer was Grace Hopper who wrote the very first compiler for the programming language A-0.[47] Another known part of computer history heavily influenced by woman is the ENIAC project. It was one of the first large-scale digital computer designed primarily to calculate artillery firing tables and is told to have done more arithmetic than the whole human race prior to 1945 (Feller & Fitzgerald 2002: 27). ENIAC was programmed mainly by women known as the ENIAC girls (Fritz 1996).

Age

In the WIDI study the median age of developers is 25 while the arithmetic mean is 27 (Robles et al. 2001a: 24-25 and own computation). Roughly 10% of the developers are below 20 years old, 62% between 20 and 29 years, 23% between 30 and 39 years, 4% between 40 and 49 years, and 1% is older than 50 years.

In the FLOSS study subjects were on the average 27 years old (mean) with a median at 26 years (Ghosh et al. 2002a: 8). Only 25% of subjects were older than 30 years that is similar to the WIDI study with 28% for the same range. In the FLOSS-US study subjects were 27 years (median) old and 50% of subjects were between 23 and 33 years old (David, Waterman, & Arora 2003b: 16). In the FLOSS-JP study developers were on average (arithmetic mean and median) 31 years old (Mitsubishi Research Institute 2004b). The arithmetic mean of developers in the FLOSS-Asia study was 28 years with a median of 27 (Mitsubishi Research Institute 2004a).

Subjects of the BCG study were on average 30 years (mean, Lakhani & Wolf 2005: 8-9) and subjects of the study of Hertel et al (2003: 1167) were 30 years (mean) old.

Average age								
	FLOSS	FLOSS-US	FLOSS-JP	FLOSS-Asia	WIDI	BCG	Hertel et al.	Luthiger
Median	26	27	31	27	25	[a]	[a]	27
Mean	27.1	[a]	31.2	27.9	26.8	30	30	28.7
N	2774	1588	547	138	5487	684	141	1330

[a] not reported

Figure 5: Comparison of average age in different FOSS studies [48]

[46] In honor of Augusta Byron Ada was used as the name of a programming language of the U.S. Defense Department. Ada is primarily used for highly reliable and complex applications such as the avionics applications in the Airbus 320, the Swiss railroad train control, or the electronic funds transfer of the Swiss Post (ISO 1998).

[47] Grace Hopper is also known for making the term “computer bug” popular although she had not invented it. She fixed in 1945 a moth to a computer log with the entry: "First actual case of bug being found" (Naval Historical Center 1999).

[48] David et al. (2003b: 16), Ghosh et al. (2002a: 8), Hertel et al. (2003: 1167), Lakhani and Wolf (2005: 8-9), Luthiger (2006: 183-184), Mitsubishi Research Institute (2004a), Mitsubishi Research Institute (2004b), and Robles et al. (2001a: 24-25).

Nationality and residence

Some studies struggled hard to attract subjects from all over the world by announcing the survey in different languages and on different news sites and mailing lists (e.g. Ghosh et al. 2002c) while others did not. Depending on these procedures the studies seem to have a different composition of subjects. Nevertheless, if the five studies presented below are compared one may see some patterns that are astonishingly clear.

Probably the most striking pattern is the low participation rate of Asian developers (see Figure 6). In all five studies there are less Japanese than Danish developer although Japan (JP) has approximately 25 times more inhabitants than Denmark (DK, CIA 2007d). Interestingly, there is no paper known that addresses this puzzle. Therefore, it will be discussed in length below.

Nationality of survey subjects

Country	Fraction of subjects in percent					Population in million
	WIDI	BCG	FLOSS	FLOSS-US	Luthiger	
Brazil	1.39%	1.34%	0.72%	1.14%	1.86%	178.5
Denmark	1.35%	0.89%	1.54%	1.41%	0.93%	5.4
France	5.26%	3.71%	16.30%	3.75%	5.27%	60.1
Germany	21.28%	11.44%	12.27%	23.09%	18.51%	82.5
Hong Kong	0.04%	0.15%	0.00%	0.13%	0.00%	7.1
Italy	0.80%	2.23%	7.84%	2.88%	3.25%	57.4
Japan	0.07%	0.15%	0.14%	0.27%	0.00%	127.7
Poland	0.81%	0.30%	0.18%	2.68%	1.78%	38.6
South Korea	0.02%	0.15%	0.00%	0.00%	0.00%	47.7
Sweden	2.32%	2.23%	3.49%	2.01%	3.49%	8.9
Switzerland	1.80%	1.49%	1.77%	1.81%	3.10%	7.2
Taiwan	0.04%	0.15%	0.00%	0.33%	0.00%	22.3
United Kingdom	4.63%	6.69%	6.52%	3.95%	4.65%	59.3
United States	28.60%	39.67%	10.37%	21.55%	19.21%	294.0
N	**5399**	**673**	**2208**	**1494**	**1291**	

Figure 6: Fraction of subjects of selected countries in different FOSS studies [49]

One explanation for the low participation rate of Asian developers may be culture respectively the low Asian scoring on the individualism scale of Hofstede (74 in DK, 46 in JP, Hofstede 1980: 315).[50] It is argued here

[49] CIA (2007d), David et al. (2003a), Ghosh et al. (2002c), Lakhani and Wolf (2002: 22), Luthiger (2006: 76-77), and own computation of WIDI data.

[50] Apart from individualism another aspect of Asian culture may influence participation in FOSS development. At least in Japan it is according to the Haragei principle demanded to communicate implicitly and to strive for harmony of opinions (Davies & Ikeno 2002: 103-106). It is expected that others read between the lines and that bad news are communicated positively (De Mente 2003: 81-82). In FOSS development respectively in peer review the tone is mostly very rough. If something is considered not good enough it is communicated very directly. Therefore Asian may do not feel comfortable. Alan Cox (Linux Format 2005) described this problem in an interview in the following way: “People from certain cultures will be very blunt, and people from other cultures don't respect that bluntness. And there are different things that really upset certain people. I've had code from someone in Japan which I was absolutely sure was total garbage but what you don't do to that person is publicly say on the kernel lists ‘this is crap’, which is what they do

that FOSS communities stress very strongly personal achievements and individual interests. An expression of the importance of personal achievement in FOSS development is meritocracy (see chapter 4.3.7). A developer will only be respected and accepted as a leader if he writes impressive source code. Participation in FOSS development is also highly driven by individual interests. Developers choice themselves to what project they contribute (see chapter 4.3.2). They have always an exit option and they may leave the project any time. It is therefore argued that FOSS communities show strong characteristics of an individualistic culture (see chapter 5.4.2.1.2).

Yamagishi (1988b; 1994) showed impressively how cooperation of subjects from collectivist but not from individualistic countries collapses in social dilemmas if sanctions are removed. Additionally, subjects from collectivistic countries opt more often for an exit option in a social dilemma than subjects from individualistic countries (Yamagishi 1988a). The reason for this may be that strong sanctioning in collectivistic societies i) undermines intrinsic motivation and that ii) it makes individuals less trustful of others. As a consequence, subjects from a collectivistic society opt more often for the exit option (=do not participate in FOSS development) if they are confronted with the individualistic task of participation in FOSS projects.

However, there may be also other factors that influence the number of subjects in FLOSS development (exemplary data for DK and JP are presented):

- ***Number of inhabitants*** (5 million in DK, 128 million in JP, CIA 2007d): The more inhabitants a country has the more potential subjects may participate in FOSS development. Therefore, the number of inhabitants of a country should be positively related to their fraction in FOSS development.
- ***Fraction of inhabitants with access to the Internet*** (69% in DK, 68% in JP, CIA 2007b): To participate in FOSS development access to the Internet is necessary.[51] Therefore, one may suggest that the fraction of subjects with access to the Internet is positively correlated to participation in FOSS development.
- ***Unemployment rate*** (4.5% in DK, 4.1% in JP, CIA 2007e): Time is needed to participate in FOSS development. Because time is a scare resource especially to the ones working full-time one may argue that the unemployment rate of a country is positively related to participation in FOSS studies.
- ***GDP per capita*** (37’100$ in DK, 33’100$ in JP, CIA 2007a): To state how GDP affects participation in FOSS development is not trivial. On the one hand there is decreasing utility for money. Therefore one may assume that the participation of fun seekers (see chapter 3.2.1) who are motivated by the pleasure of writing source code is positively related to GDP per capita. If GDP per capita is high a fun seeker will find it more rewarding to spend another hour on FOSS development and not on earning money. On the other hand investors in human capital and investors in reputation (see chapter 3.2.2.3) will more likely contribute to FOSS development if the GDP per capita is low. Compared to other means of corresponding investments in human capital or reputation the resources needed to participate in FOSS development are, apart from time, very low. In consequence, participation in FOSS development may be attractive for poorer individuals.

to a lot of people. Instead I mailed him back with a list of things I thought were problems. I said, ‘This is an interesting prototype, I look forward to receiving the real thing.’"

[51] Of course access to computer hardware and software is also necessary. However, one can assume that most subjects having access to the Internet also have access to computer hardware and software suitable for software development.

- ***English language*** (Danish is a Germanic language while Japanese is not, Voegelin & Voegelin 1977: 139-146): Almost all relevant FOSS development is conducted in English. One may therefore assume that subjects with a native language other than English will have more problems to participate in FOSS development than a native speaker. Similarly, subjects speaking a Germanic language (e.g. Dutch or Swedish) will learn English faster than subjects speaking a non-Germanic language (e.g. Russian, Chinese, or Swahili). It can be assumed that the native language will be related to participation in FOSS development.
- ***Working hours*** (1688.9 annual hours in DK, 1842.0 in JP, ILO 2002: 219-224): It seems to be reasonable to assume that in countries with long working hours subjects will find hardly time to contribute to FOSS development. It can therefore be suggested that working hours are negatively related to FOSS participation.

If Denmark and Japan are compared the fraction of inhabitants with access to the Internet, the unemployment rate, and the GDP per capita are subjectively very similar. It can be therefore assumed that these factors are hardly responsible for the different participation rate in FOSS development. The number of inhabitants in Denmark is considerably lower than in Japan and can therefore not explain the observed participation patterns. However, Dansk is unlike Japanese a Germanic language. Additionally, Japanese individuals have longer working hours and are less individualistic. These characteristics probably explain the difference between Japanese and Danish involvement in FOSS development.

To research the question of participation in FLOSS development a series of linear regressions with stepwise selection of variables were run.[52] The dependant variable is the number of subjects in the corresponding study.[53] As independent variables the number of inhabitants of a country (CIA 2007d)[54], the fraction of inhabitants with access to the Internet (CIA 2007b), the unemployment rate (CIA 2007e), GDP per capita (CIA 2007a), the dominant language (CIA 2007c)[55], the annual working hours (ILO 2002: 219-224)[56], individualism (Hofstede 1980: 315),[57] power distance (Hofstede 1980: 315), masculinity (Hofstede 1980: 315), and uncertainty avoidance (Hofstede 1980: 315) were introduced.

---

[52] In stepwise selection at each step the independent variable that has the smallest probability of F is entered into the regression provided that it is below 0.05 and that it has not already been entered into the equation (SPSS 2005: 430). Variables already in the regression equation are removed if their probability of F is above 0.1. This procedure terminates if there are no more variables eligible for inclusion or removal (see also Broersen 1986; and Whittingham et al. 2006).

[53] Some studies make a distinction between country of residence and nationality (David, Waterman, & Arora 2003b: 15; Ghosh et al. 2002a: 18-20; Robles et al. 2001a: 22-24). In these cases the country of residence was included in the computation. The reason for this is that culture is dominated by the corresponding institutions that are in turn tightly related to the country of residence (see the institutional view of culture in chapter 5.4.2.1.2).

[54] The number of inhabitants is given in millions.

[55] This variable is dummy-coded. English speaking countries are coded as 1 and non-English speaking countries as 0. A country is counted as an English speaking country if English is an official language, if more than 30% speak English, or if it is mentioned as the first language in the CIA factbook (CIA 2007c).

[56] The annual working hours refer to the year 1999 except for Canada and France (both 1998). For Austria, China, India, Indonesia, Israel, Poland, Singapore, South Africa, and Taiwan no data from the same source could be found. Other databases (e.g. OECD) that include figures for some but not all of the mentioned countries have the problem that data for other countries are missing. The data from different sources was not mixed due to methodological reasoning.

[57] Concerning cultural orientation replications of the original research of Hofstede (2005: 25-28) showed similar results for non-IBM employee's. These replications also suggest that the cultural orientation of the surveyed countries did not change since the original study (Hofstede & Hofstede 2005: 28).

It is hypothesized that in the linear regressions i) individualism is so important that it is included by the stepwise selection method (respectively that the F value is below 0.05) and that there is ii) a recognizable change of the adjusted R square.

Country participation in the BCG survey											
Model		***Coefficients***					***Model summary***				
		Unstandardized		Standardized	T	Significance	R	R Square	Adjusted R Square	Std. Error of the estimate
		B	*Standard error*	*Beta*						
1	*Constant*	-4.169	9.806		-0.425	0.675	0.768	0.590	0.570	37.495
	Inhabitants	0.611	0.114	0.768	5.369	0.000				
2	*Constant*	-68.744	19.959		-3.444	0.003	0.867	0.752	0.726	29.938
	Inhabitants	0.596	0.091	0.749	6.551	0.000				
	Individualism	1.054	0.300	0.402	3.517	0.002				

Excluded in model 2: Internet users, annually working hours, GDP per capita, language (dummy), power distance, masculinity, uncertainty avoidance, unemployment
Method of variable selection: stepwise

Figure 7: Stepwise linear regression: Participation in the BCG study

Country participation in the Luthiger survey										
Model		***Coefficients***					***Model summary***			
		Unstandardized		Standardized	T	Significance	R	R Square	Adjusted R Square	Std. Error of the estimate
		B	*Standard error*	*Beta*						
1	*Constant*	13.665	14.174		0.964	0.347	0.625	0.391	0.361	54.200
	Inhabitants	0.590	0.165	0.625	3.585	0.002				
2	*Constant*	-59.819	32.226		-1.856	0.079	0.735	0.540	0.492	48.338
	Inhabitants	0.573	0.147	0.607	3.898	0.001				
	Individualism	1.200	0.484	0.386	2.479	0.023				

Excluded in model 2: Internet users, annually working hours, GDP per capita, language (dummy), power distance, masculinity, uncertainty avoidance, unemployment
Method of variable selection: stepwise

Figure 8: Stepwise linear regression: Participation in the Luthiger study

Country (residence) participation in the FLOSS survey										
Model		***Coefficients***					***Model summary***			
		Unstandardized		Standardized	T	Significance	R	R Square	Adjusted R Square	Std. Error of the estimate
		B	*Standard error*	*Beta*						
1	*Constant*	44.327	23.707		1.870	0.076	0.448	0.201	0.161	90.65099
	Inhabitants	0.618	0.275	0.448	2.244	0.036				
2	*Constant*	-66.561	55.497		-1.199	0.245	0.600	0.360	0.293	83.242
	Inhabitants	0.592	0.253	0.430	2.339	0.030				
	Individualism	1.811	0.834	0.399	2.172	0.043				
3	*Constant*	-233.650	81.579		-2.864	0.010	0.728	0.531	0.452	73.24908
	Inhabitants	0.564	0.223	0.410	2.530	0.021				
	Individualism	2.708	0.813	0.597	3.330	0.004				
	Unemployment	15.849	6.199	0.458	2.557	0.020				
4	*Constant*	-440.104	118.540		-3.713	0.002	0.798	0.637	0.551	66.31918
	Inhabitants	0.653	0.206	0.474	3.174	0.006				
	Individualism	1.691	0.866	0.373	1.952	0.068				
	Unemployment	28.584	8.013	0.826	3.567	0.002				
	Internet user	3.139	1.410	0.605	2.227	0.040				

Excluded in model 4: Annually working hours, GDP per capita, language (dummy), power distance, masculinity, uncertainty avoidance
Method of variable selection: stepwise

Figure 9: Stepwise linear regression: Participation in the FLOSS study

Country (residence) participation in the _FLOSS-US survey											
Model		***Coefficients***					***Model summary***				
		Unstandardized		Standardized	T	Significance	R	R Square	Adjusted R Square	Std. Error of the estimate	
		B	*Standard error*	*Beta*							
1	*Constant*	8.374	21.739		0.385	0.704	0.613	0.376	0.344	83.124	
	Inhabitants	0.875	0.252	0.613	3.469	0.002					

Excluded in model 1: Internet users, annually working hours, GDP per capita, language (dummy), power distance, masculinity, uncertainty avoidance, unemployment, individualism
Method of variable selection: stepwise

Figure 10: Stepwise linear regression: Participation in the FLOSS-US study

Country (residence) participation in the WIDI survey										
Model		***Coefficients***					***Model summary***			
		Unstandardized		Standardized	T	Significance	R	R Square	Adjusted R Square	Std. Error of the estimate
		B	*Standard error*	*Beta*						
1	*Constant*	7.972	77.963		0.102	0.920	0.717	0.514	0.490	298.114
	Inhabitants	4.167	0.905	0.717	4.603	0.000				
2	*Constant*	-436.527	171.145		-2.551	0.020	0.811	0.658	0.622	256.708
	Inhabitants	4.063	0.780	0.699	5.207	0.000				
	Individualism	7.258	2.571	0.379	2.824	0.011				

Excluded in model 2: Internet users, annually working hours, GDP per capita, language (dummy), power distance, masculinity, uncertainty avoidance, unemployment
Method of variable selection: stepwise

Figure 11: Stepwise linear regression: Participation in the WIDI study

As can be seen in Figure 7, Figure 8, Figure 9, and Figure 11 individualism is included in the linear regression to explain participation in the BCG study, the Luthiger study, the FLOSS study, and in the WIDI study. The change of the R square is considerable. Individualism explains between 14% and 16% of the variance in the data and seems therefore to be an important factor for participation in FOSS development. In the FLOSS-US study individualism is not included and therefore does not contribute anything significant to the regression (see Figure 10).

A variable included in all linear regressions is the number of inhabitants of a country. The more inhabitants a country has the more developers participate in FOSS development. In the FLOSS study the unemployment rate and the fraction of citizens having access to the Internet are also included in the regression. In both cases the relationship to participation in FOSS development is positive.

If the algebraic signs of the beta coefficients of the excluded non-significant variables are counted one can vaguely guess that:

- working hours are negatively,
- unemployment is positively,
- per capita GDP is positively,
- power distance is negatively,
- masculinity is negatively,
- uncertainty avoidance is negatively, and that
- the fraction of citizens having access to the Internet is positively related to FOSS participation.

In conclusion, it can be assumed that next to the number of inhabitants of a country its degree of individualism is crucial for the FOSS participation rate. Very astonishingly, language does not seem to play a significant role for participation in FOSS development.[58]

Migration of developers

In some FOSS studies an interesting distinction between nationality and country of residence has been made (David, Waterman, & Arora 2003b: 15; Ghosh et al. 2002a: 18-20; Robles et al. 2001a: 22-24). With the help of these two variables migration patterns of developers can be observed. As can be seen in Figure 12 there is an extreme migration of FOSS developers to the United States. Other countries with a positive net migration are Germany and Switzerland while France, Poland, and India seem to have a negative net migration.

Migration of FOSS developer

Country[a]	Difference between country of residence and nationality in percent[b]				
	WIDI	FLOSS	FLOSS-US	All[c]	
United States	5.68%	2.90%	3.85%	12.43%	Positive net migration
Germany	0.03%	0.76%	4.11%	4.90%	
Switzerland	0.41%	0.57%	0.15%	1.13%	
Denmark	0.08%	-0.11%	0.04%	0.01%	Neutral net migration
Czech Republic	-0.03%	-0.04%	0.08%	0.01%	
Brazil	0.06%	-0.01%	-0.05%	0.00%	
France	-0.41%	-0.31%	0.23%	-0.48%	Negative net migration
Poland	-0.45%	0.01%	-0.22%	-0.65%	
India	-0.41%	-0.37%	0.08%	-0.69%	

[a] This a unsystematic selection of countries.
[b] The fraction of developers with a certain nationality in a survey is subtracted from the fraction of developers living in a certain country. Positive values indicate a positve net migration to this country and negative values a negative migration away from this country.
[c] The differencenes of the WIDI, the FLOSS, and the FLOSS-US survey are added.

Figure 12: Developer migration in different FOSS studies[59]

Education

It is thought that it is very difficult to properly research the education of FOSS developers (see chapter 10.3). The studies reviewed here as well as the own study are considered to not having completely mastered this difficulty. One should be therefore careful interpreting the results below.

Depending on the study 63% to 85% of subjects have completed some sort university education (see Figure 13). This suggests that FOSS project members have on average a very high education.[60]

[58] A subsequent study could include a more sophisticated measure of the linguistic relationship to English. However, doing so is not trivial.
[59] David et al. (2003a), Ghosh et al. (2002c), and own computation of WIDI data.
[60] The fraction of individuals in the European Union holding a PhD degree is for example by far lower (Eurostat & Meri 2007: 4).

Highest completed education					
	FLOSS	FLOSS-US	FLOSS-JP	FLOSS-Asia	WIDI
Below bachelor / college	29.6%	19.4%	31.2%	15.0%	37.2%
Bachelor / college	32.6%	38.0%	35.2%	63.9%	17.4%
Master	28.1%	36.7%	27.1%	15.8%	40.6%
PhD	9.8%	5.9%	6.5%	5.3%	4.8%
Some kind of university degree	70.4%	80.6%	68.8%	85.0%	62.8%
N	2143	1498	542	133	4872

Figure 13: Highest completed education in different FOSS studies [61]

Professional background

In the FLOSS study 83.6% of the developers are employed in the IT industry or their university studies are related to IT (Ghosh et al. 2002c).[62] Most developers included in the survey are software engineers (33%), followed by IT students (16%), by programmers (11%), by IT consultants (10%), by individuals who have an IT related job at a university (5%), and by IT executives (3%).[63] Only a minuscule fraction of FOSS developers have an occupation related to IT marketing or IT product sales.

In the FOSS-JP study the professional background of 72.0% of the subjects is related to IT (Mitsubishi Research Institute 2004d). In contrast to the FLOSS study there are less IT students, less IT consultants but more software engineers. In the FOSS-Asia study 84.9% of the developers have, similarly to the FLOSS study, an IT profession or their studies are related to IT (Mitsubishi Research Institute 2004c). In the WIDI study 79.7% of the developers are professionally related to IT (Robles et al. 2001a: 31). In contrast to the FLOSS study there are less software engineers but more programmers and IT students in the WIDI study.

In the WIDI study there are 30% (own computation)[64] and in FLOSS-US study there are 29% students (David, Waterman, & Arora 2003b: 10). The FLOSS study (21% students, Ghosh et al. 2002c)[65], the BCG study (20% students, Lakhani & Wolf 2005: 9), the FLOSS-Asia study (15.8% Mitsubishi Research Institute 2004c), and the FLOSS-JP study (14.4% students, Mitsubishi Research Institute 2004d) locate fewer students.

[61] David et al. (2003a), Ghosh et al. (2002c), Mitsubishi Research Institute (2004c), Mitsubishi Research Institute (2004d), and Robles et al. (2001a: 33).

[62] In the BCG study 58% of the subjects were related to the IT industry (Lakhani & Wolf 2005: 8). In the FOSS-US study 78% of the subjects earned income from business firms related to IT (David, Waterman, & Arora 2003a). However, both results are not directly included in the presentation of results because the used methodologies are not completely comparable. In the BCG study IT students are not counted as having a relationship to IT respectively it is not distinguished between IT and non-IT students.

[63] According to the Bureau of Labor Statistics (BLS 2001) computer programmers as well as software engineers belong to the category of computer specialists.

[64] The reported 33% in the WIDI study also includes professors, assistants, and the like (Robles et al. 2001a: 32).

[65] In the FLOSS study there is another item in which subjects could indicate that they are a student (Ghosh et al. 2002c). However, the raw data for this item seems to be corrupted because they neither add up to 100% nor do they correspond to the figures in the final report (Ghosh et al. 2002a). Consequently, this data is not examined here.

Professional background				
Professional background	FOSS	FOSS-JP	FOSS-ASIA	WIDI
Non IT	16.41%	28.00%	15.10%	20.31%
Software engineer	33.10%	41.10%	27.80%	25.37%
Programmer	11.16%	10.00%	17.30%	17.44%
Consultant (IT)	9.91%	2.40%	6.00%	9.68%
Manager (IT)	3.14%	2.20%	6.00%	0.00%
Exective (IT)	0.18%	2.00%	3.80%	4.16%
Marketing / Product sales (IT)	0.09%	0.00%	0.00%	0.00%
University / Research Institute (IT)	5.03%	4.30%	9.00%	2.04%
Student (IT)	15.81%	6.50%	10.50%	21.00%
Other (IT)	5.16%	3.50%	4.50%	0.00%
N	2169	540	133	5333

Figure 14: Professional background in different FOSS studies[66]

Income

In the FLOSS study the estimated median income is roughly 2000 US$ (Ghosh et al. 2002a: 15). In the FLOSS-JP study as well as in the WIDI study the estimated median is roughly 3000 US$ (Mitsubishi Research Institute 2004d; Robles et al. 2001a: 37).

[66] Ghosh et al. (2002c), Mitsubishi Research Institute (2004d), Mitsubishi Research Institute (2004c), Robles et al. (2001a: 31).

Monthly gross income					
FLOSS-Japan			*FLOSS*		
Monthly gross income[a]	Frequency	Cumulative Frequency	Monthly gross income[b]	Frequency	Cumulative Frequency
0 US$	11.99%	11.99%	0 € / US $	7.29%	7.29%
≈ 911 US$	7.30%	19.29%	>1000 € / US $	22.32%	29.61%
≈ 1822 US $	7.87%	27.15%	1001 - 2000 € / US $	22.22%	51.83%
≈ 2734 US$	13.11%	40.26%	2001 - 3000 € / US $	18.36%	70.19%
≈ 3645 US$	16.10%	56.37%	3001 - 4000 € / US $	11.55%	81.74%
≈ 4556 US$	17.04%	73.41%	4001 - 5000 € / US $	7.10%	88.84%
≈ 6835 US$	20.22%	93.63%	5001 - 7500 € / US $	6.04%	94.88%
≈ 9113 US$	3.93%	97.57%	7501 - 10'000 € / US $	2.17%	97.05%
< 9113 US$	2.43%	100.00%	<10'000 € / US $	2.95%	100.00%

[a] In the FLOSS-JP survey no range of income was provided. Instead, subjects could choose the value that fits best. This inaccurancy seems to be indicated by the ≈ sign.

[b] The first two categories are not mutually exclusive. However, this most probably did not produce a major bias.

Figure 15: Monthly gross income in different FOSS studies[67/68/69/70]

Payment for contribution

There are sharp differences between FOSS studies whether developer are paid or not for their contribution (see Figure 16).[71] There are differences between 13% and 50% of paid developers. This makes the interpretation of the corresponding results very difficult respectively meaningless.

Study	Percentage of paid developers
BCG (Lakhani & Wolf 2005: 9)	13%
Hars and Ou (2002: 31)	16%

[67] Ghosh et al. (2002c) and Mitsubishi Research Institute (2004d).

[68] The FLOSS-Asia study has also surveyed income (Mitsubishi Research Institute 2004c). However, there seems to be a defect in the corresponding item. Therefore, the corresponding data is not presented here.

[69] The reported gross income in the FLOSS-JP study is presented in Yen and per year (Mitsubishi Research Institute 2004b). To compare the results the data was divided by 12 and transformed to US$ (average exchange rate of September and October 2003, Bank of Japan 2006).

[70] Subjects answered the WIDI study between June 24 and August 14, 2001 (Robles et al. 2001a: 18). The WIDI study assumed dollar-euro parity (Robles et al. 2001b) while actually one euro was exchanged between 0.8607 and 0.8968 US$ (ECB 2006) .The FLOSS study started in February 2002 and ended in early April 2002 (Ghosh et al. 2002a: 5). It assumes also dollar-euro parity. Actually, one euro was exchanged for 0.8632 US$ at February 1, 2002 and for 0.8786 at April 2 (ECB 2006). The approximate exchange rate in this study was 1.25 US$ for one euro.

[71] Note that for other items such as “developer’s age” there is very good concordance.

WIDI (Robles et al. 2001a: 33)	21%
FLOSS-JP (Mitsubishi Research Institute 2004d)	27%
FLOSS-US (David, Waterman, & Arora 2003a)	41%
FreeBSD (Jorgensen 2000, 2001)	43%
Linux kernel (Hertel, Niedner, & Herrmann 2003: 1168)	43% (20% paid and 23% sometimes paid)
FLOSS-Asia (Mitsubishi Research Institute 2004c)	45%
FLOSS (Ghosh et al. 2002c)	50%

Figure 16: Percentage of paid developers in different FOSS studies

It is argued here that the huge differences are due to survey construction, methodology, question wording, and sample effects. It must be stated that deficient and imprecise wording seems to be a major problem for researching the extent to which developers are paid:

- Many studies do not distinguish between full and partial payment. A developer paid to maintain a Linux driver but who also contributes to other parts of the Linux kernel in spare time may or may not agree to the question whether he is paid for FOSS participation.
- In the BCG study 13% of developers report receiving a direct payment for project participation while at the same time 54% of FOSS project members report that they contribute to FOSS projects during working hours (Lakhani & Wolf 2005: 9-10). Therefore, there are cases in which developers contribute to FOSS with or without consent of the employer. The status of being paid applies to both cases. Many studies do not distinguish between developers paid with or without the consent of the employer. However, it seems to be clear that these two possibilities should not be mixed if a meaningful interpretation should be made.

It can be concluded that research on the question whether developers are paid or not is not yet exhausted. Further studies have certainly to be more accurate in wording.[72]

Project size

An empirical investigation of Krishnamurthy (2002) showed that many FOSS projects have only a few active developers. In the 100 most active projects at SourceForge with status mature the average number of *developers* was 7 (see Figure 17).[73]

[72] Unfortunately, this study does not completely master the discussed problems.

[73] The largest FOSS projects are not hosted at SourceForge. Surveying only this hosting platform introduces therefore clearly a bias.

Descriptive Statistics of Developers and Project Administrators at SourceForge						
	Project administrators			***Developers***		
	2002	2004	2007	2002	2004	2007
Arithmetic mean	2.21	2.44	2.88	6.61	13.17	21.02
Median	1	2	2	4	6	11
Mode	1	2	2	1	6	8
Minimum	1	1	1	1	1	1
Maximum	14	8	15	42	103	88
Standard deviation	1.91	1.69	2.64	8.24	19.89	21.02

Figure 17: Descriptive statistics of FOSS projects at SourceForge[74]

Based on the study of Krishnamurthy (2002) a similar study was carried out in 2004 and 2007.[75] The average number of developer increased to 13 developers on average in 2004 and to 21 developers on average in 2007.

Contributions: Spent time

As can be seen in Figure 18 approximately one-quarter of the subjects spend less than two hours a week for their contributions to FOSS projects. Another quarter of the subjects spend between two and five hours a week. A minority of 3% to 7% spend more than forty hours a week.

Hours of work per week	Frequency FLOSS	Frequency FLOSS-US	Frequency FLOSS-JP	Frequency FLOSS-Asia	Frequency WIDI
> 2 hours	22.51%	45.58%	36.50%	22.60%	34.61%
2-5 hours	26.13%		25.20%	27.10%	
6-10 hours	20.89%	23.80%	15.80%	18.00%	31.61%
11-20 hours	14.26%	16.63%	10.40%	15.80%	18.50%
21-40 hours	9.14%	10.86%	6.90%	10.50%	9.96%
< 40 hours	7.07%	3.06%	5.20%	6.00%	5.33%
N	2461	1588	547	138	5233

Figure 18: Spent time for FOSS development in different FOSS studies [76]

[74] Krishnamurthy (2002) and SourceForge (2004a; 2007).

[75] The study was in two aspects different to the one of Krishnamurthy. *Firstly*, only the twenty most active and the eightieth to hundredth most active projects were analyzed. *Secondly*, not only mature but all SourceForge projects were included. As can be seen in Figure 33 the status mature applies only to few projects and is therefore not very representative for FOSS projects in general. Development of mature projects is probably in many cases finished which in turn influences the number of developers in the project. Due to these methodological differences the results of the own two studies in 2004 and 2007 and the study of Krishnamurthy (2002) are not completely comparable. However, it is thought that the methodological differences are not so big that deliberate comparisons are impossible.

In the FLOSS-US study the subjects worked on average 11 (mean) respectively 7 (median) hours a week for the project (David, Waterman, & Arora 2003b: 36). In the BCG study subjects spent 8 (mean) respectively 3 (median) hours on their focal project and 14.1 hours (mean) respectively 10 hours (median) on FOSS projects in total (Lakhani & Wolf 2005: 10). In the study of Hertel et al. (2003: 1168) subjects contributed on average 18.4 hours a week (developer group) or 6.6 hours (interested readers) on Linux kernel development. Luthiger (2006: 78) found in his study that developer spend 13 hours (mean) respectively 8 hours (median) for FOSS development.

Contributions: Asymmetrical distribution

The Orbiten Free Software Survey was one of the first major studies on participation in FOSS development. The researchers studied 3149 FOSS projects with approximately 25 million lines of code written by 12’706 developers (Ghosh & Prakash 2000). They found that the top ten percent of actors account for 72% of submitted code and the top twenty percent account for 81%. This suggests that there are different types of developers in FOSS projects. It may be that in a project there are core developers who contribute a lot, advanced users who fix bugs, and regular users who reports bugs.

Contributions: Individuals versus organizations

In a study commissioned by the European Commission it was found that about two-thirds of the source code is contributed by individuals (UNU-MERIT et al. 2006: 18). In the year 2005 62% of the contributed source code of the Debian project was written by individuals, 19% by companies, 6% by universities, 8% by foundations, and the rest is unknown or by development groups (UNU-MERIT et al. 2006: 50).

In the Orbiten Free Software Survey by far the most important contributor to FOSS projects is an organization, namely the FSF (Ghosh & Prakash 2000).

Topic

Most FOSS projects are related to the Internet (see Figure 19). The four most frequent topics (Internet, software development, system, communications) of FOSS development account for more than half of all FOSS projects.

[76] David et al. (2003a), Ghosh (2002c), Mitsubishi Research Institute (2004a), Mitsubishi Research Institute (2004b), and Robles et al. (2001a: 37).

Topic	N	in %
Internet	23'883	16.23
Software Development	20'978	14.26
System	19'273	13.10
Communications	14'835	10.08
Games / Entertainment	14'023	9.53
Multimedia	12'558	8.54
Scientific / Engineering	11'139	7.57
Office / Business	6'956	4.73
Database	5'840	3.97
Education	3'646	2.48
Desktop Environment	3'056	2.08
Security	2'730	1.86
Other	8'208	5.58
Total	147'125	100

Figure 19: FOSS Topics at SourceForge[77]

2.5 Licenses

Intellectual property may be protected by trade secrets, trademarks, patents[78] and in the case of software particularly by copyrights. A license is the permission to execute specified rights granted to another party by the holder of the intellectual property (Messerschmitt & Szyperski 2003: 267-284). Licenses are very central in FOSS development.[79] They are the key to "lower the barriers to cooperation" and an important part of the phenomenon (O'Reilly 1999: 37). The FSF (2006e) classifies licenses according to four key questions:

i. Does it qualify as a free software license?[80]

ii. Is it compatible with the GNU GPL?

iii. Does it cause any particular practical problems?

iv. Is it a copyleft license?

The question whether a license qualifies as free is related to the free software respectively open source definition that has already been discussed in chapter 2.1. Compatibility with the GNU GPL is for practical reasons highly important.[81] If a software license is incompatible with the GNU GPL source code can hardly be reused which is a big advantage of the FOSS production model. However, for this dissertation the question of

[77] SourceForge (2006c).

[78] Software patents such as Amazon's One-Click Shopping patent (Hartman et al. 1999) are controversially discussed (see also Luethi 2005). Inter alia it is criticized that the software industry changes faster than the patent system can accommodate (Messerschmitt & Szyperski 2003: 269). In a highly regarded paper Bessen and Hunt (2004) argue that software patents in fact lowered R&D intensity of the software industry.

[79] For an excellent review on legal issues and licenses see ifrOSS and Jaeger (2005). For an overview in English see McGowan (2005).

[80] A license considered as a free software license does also apply as an open source license and vice versa (see 2.3.2).

[81] The degree it is thought that compatibility with the GNU GPL is important seems to be influenced by culture. In both the FLOSS-Asia and the FLOSS-JP study the following question was asked "Which license do you choose, if your code never depends on any other source code?" In the FLOSS-Asia study (Mitsubishi Research Institute 2004c) 65% of subjects stated that they would choose a GNU GPL compatible license while in the FLOSS-JP study (Mitsubishi Research Institute 2004d) only 42% did so.

compatibility is of minor relevance. This also applies to particular practical license problems. Such problems may be important for practical purposes but they are not relevant for this study. The following subchapter therefore deals mainly with the distinction between free / non-free and copyleft / non-copyleft licenses.

2.5.1 Typology of FOSS licenses

In the following, the typology presented in Figure 20 is discussed. At the outset a distinction between *copyrighted* and *non-copyrighted* FOSS software is made. Software in the public domain is non-copyrighted and discussed as a special case of FOSS software. For copyrighted software a distinction between *copyleft* and *non-copyleft* licenses is made. Copyleft licenses are differed further in *weak* and *strong* copyleft licenses.

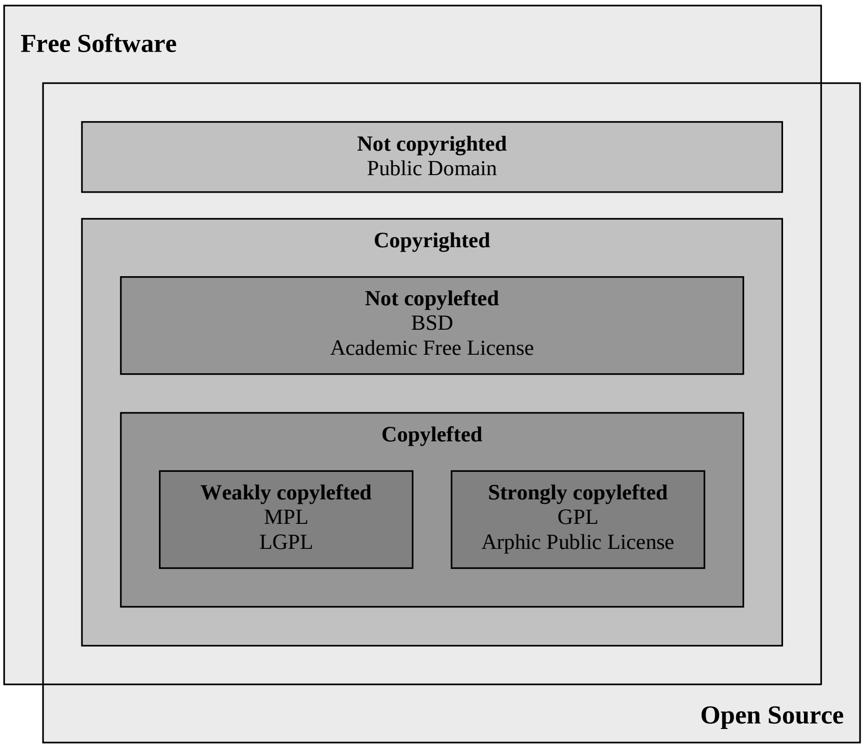


Figure 20: Typology of FOSS licenses[82]

[82] Figure 20 is inspired from the known classification of Chao-Kuei (without year). However, the classification was altered in several issues. Proprietary software licenses and its classification are omitted because such software is not the topic of this study. Also omitted is the XFree86 license because of two reasons. *Firstly*, the current XFree86 License 1.1 is due to changes in its terms not a free software license anymore (FSF 2006e). *Secondly*, the X server that is licensed under the XFree86 license is, partly because of discontent with the new license terms, replaced in most operating systems by X.Org or will be replaced in the near future. Therefore, the XFree86 license seems not to be of relevance anymore. In addition Chao-Kuei does not distinguish *explicitly* between copyrighted and non-copyrighted software and does not include a differentiation between strong and weak copyleft licenses. Because such differentiation is thought to be important they are added to the basic classification Chao-Kuein.

2.5.1.1 Non-copyrighted software

Public Domain

Software is in the public domain if nobody can establish or maintain a legal claim. Therefore it is not copyrighted which in turn obsoletes licenses. *Technically* speaking software in the public domain is therefore not related to licenses. However, public domain can be considered *practically* as a non-copyleft FOSS license.

There are several reasons why software is in the public domain and copyright protection does not apply. The corresponding legislation may be sharply different from country to country. In the US the work of the government is excluded from copyright law and therefore automatically in the public domain (17 U.S.C. §105). A way of putting software in the public domain is, at least in the US, to publicly distribute software "with an explicit disclaimer of copyright protection by the copyright owner" (37 C.F.R. §201.26b(3)). Copyright may also expire that puts software as a consequence in the public domain (but see the Sonny Bono Copyright Term Extension Act). Work created before the existence of copyright law such as the bible is also in the public domain. However, software has not been produced prior to the installment of the copyright law. Similarly, the expiration of copyrights does hardly apply to software.

2.5.1.2 Copyrighted software

In all countries where the Berne Convention standards do apply, one gets automatically a copyright on original software.[83] However, while no registration has to be obtained in order to get copyright protection it is in some countries an advantage doing so because it "constitute *prima facie* evidence of the validity of the copyright" (17 U.S.C. §410c).

The owner of the copyright has the exclusive right to sell copies of the software, to create derivative works, to run the software, and to sell these rights to others. The copyright holder may grant a license that specifies what rights its holder has. For proprietary software the licensee has usually to pay for the specified permissions (prominently the right to run software). FOSS licenses give everybody the permission to run, modify, and redistribute without paying for it (see also FOSS definitions).[84] Some licenses have further restrictions such as to preserve the name of the author or the license if software is modified.

Non-copyleft licenses: The BSD style

There are many different non-copylefted licenses. The most important are certainly the BSD licenses respectively its derivates that will now be discussed in depth. According to Behlendorf (1999: 164) the BSD license can be summed up as follows: "Here's this code, do what you like with it, we don't care, just give us credit if you try and sell it." A BSD style license has the advantage[85] that code from an existing project can be taken with no worries about restrictions on future use or redistribution. Proprietary source code can be mixed with source code under a BSD style license.

---

[83] Not parties to the Berne Convention are for example Angola, Iran, Nauru, or Tuvalu (WIPO 2008).

[84] The GNU GPL allows charging a substantial fee for the distribution of software. However, *after* the software is distributed a user cannot be demanded to pay again for redistribution.

[85] This may be also considered as a disadvantage (see below and chapter 2.6.1.1).

The problem of BSD style licenses is the risk that everybody wants to take but nobody wants to give back which threatens the further existence of a project (see the NetBSD example in chapter 4.2). The Apache HTTP server project historically has suffered from the problem that modifications have been taken privately: “Companies have developed technology around it {Apache} that we would have like to have seen offered back to the project” (Behlendorf 1999: 165). Behlendorf argues, however, that if Apache had been put under a copyleft license the technology would perhaps not have been produced at all.

There is a discussion going on whether BSD style or GNU GPL style licenses are more free (ifrOSS & Jaeger 2005: 3; Raymond 2004: 47). The advocates of the BSD license argue that it has fewer restrictions and is therefore freer (e.g. Brewer 2005). The advocates of the GNU GPL license make the point that the copyleft license assures that software remains free (e.g. Asay 2005). Both argumentations are correct. Which license is freer depends on the definition of freedom (Kuhn & Stallman 2006).

Copyleft licenses

FOSS licenses ensure only that a certain piece of software is free but not necessarily modified source code. Source code can, depending on the license, be put under a proprietary license. Copyleft commands additionally that *modified* and *extended* versions of software are free as well (FSF 2006f).[86] In legal terms copyleft licenses state that i) the source code of software is copyrighted, that ii) everyone has the rights to use, modify, and redistribute this source code or any program derived from it but only if iii) the distribution terms are unchanged. Because the distributions terms cannot be changed modified versions of copylefted software must also grant the specified rights. Modified software under a non-copylefted FOSS license such as the BSD license can be put under a proprietary license because a provision as specified in iii) does not exist.

The initial naming of term “copyleft” is based on an idea of Don Hopkins who wrote a letter to Richard Stallman with “copyleft - all rights reversed" written on the envelope (Stallman 2001a). Stallman built upon the word copyleft the GNU GPL.

Examples of copyleft licenses are the GNU GPL, the Common Public License, the IBM Public License, the Jabber Open Source License, the Eclipse Public License, and the NetHack General Public License.

[86] The copyleft term has been wrongly called viral. The misunderstanding seems to be based on a speech of Microsoft Senior Vice President Craig Mundie (2001). He stated that the GNU GPL has a viral aspect and claimed that the GNU GPL makes it “impossible to distribute software on a basis where recipients pay for the product rather than just the cost of distribution”. These statements must be challenged: *Firstly*, software under the GNU GPL does (and of course cannot) legally infect unrelated software as could be suggested by the term viral. The GNU GPL states that for work that “can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply” (FSF 2006c). Additionally, “aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License”. *Secondly*, one may think that work derived from software under the GNU GPL is licensed automatically under the GNU GPL. This is wrong. However, not putting the modified version under the GNU GPL would make it illegal to use the software and any rights under the GNU GPL would be terminated. A court would have to decide how this illegal use of the original software was to be sanctioned. In either case: the modified version is not automatically licensed under the GNU GPL. *Thirdly*, license terms that demand that derived work must be put under the same license may also apply for commercial licenses. If somebody gets the permission to customize commercial software on its own it may be possible that the corresponding license terms demand that the modified software cannot be distributed without paying a fee to the owner of the original software. *Fourthly*, The GNU GPL does not prevent that customers pay for software. Actually, there are commercial firms listed on the stock exchange that sell commercial Linux distributions.

Although the BSD license seems to be at first more business friendly not all agree with this reasoning. Matt Asay (2005), director of Novell's Linux Business Office, argues that “the GPL is one of the most exciting, innovative capitalist tools ever created” and that “the GPL facilitates commercial involvement in open source communities”.

Weak and strong copyleft licenses: The difference

Copyleft licenses can be further subclassified in *strong* and *weak* copyleft licenses. While strong copyleft licenses do not allow linking with non-free modules weak copyleft licenses do. To explain linking let us assume that there are two programs X and Y. For proper functioning both programs need to be able to draw rectangles (e.g. for dialog boxes). There are different possibilities to do so: *Firstly*, one can implement the task “drawing a rectangle” in the program itself as in the example of program X (see Figure 21). *Secondly*, one can implement the functionality by dynamically link to a library (e.g. the Cairo graphic library) that is able to draw a rectangle. In the example below program Y links to library Z that is probably also used by other programs (see Figure 21). *Thirdly*, it is possible to use a static library (not discussed here).

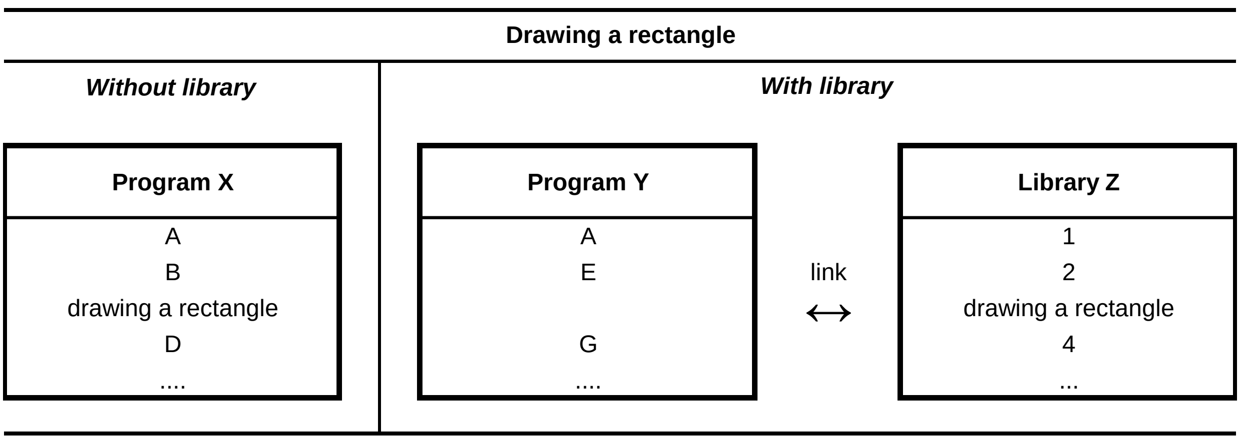


Figure 21: Implementing the task of drawing a rectangle directly or by linking to a library

Let us assume that both program X and Y are under a *strong copyleft license* such as the GNU GPL (see Figure 22). In this situation the modified program X as well as the new library Z must be put or remain under a GNU GPL license in order to use the software legally.

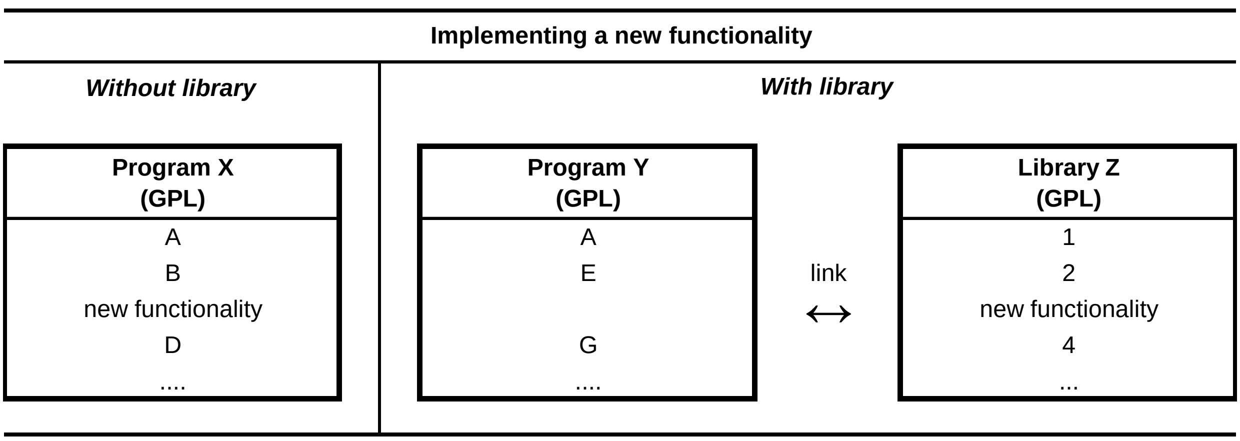


Figure 22: Adding a new functionality to GNU GPL’ed software

Let us now assume that program X and Y are under a *weak copyleft license* such as the GNU LGPL (see Figure 23). Again, the new feature may be implemented in the program itself (example of program X) or by linking to library Z (example of program Y). The modified program X must be imperatively put under a GNU LGPL license.[87] However, it is not mandatory that library Z is licensed under the GNU LGPL. Library Z may be put under a GNU GPL, a GNU LGPL, a proprietary license, or any other license.

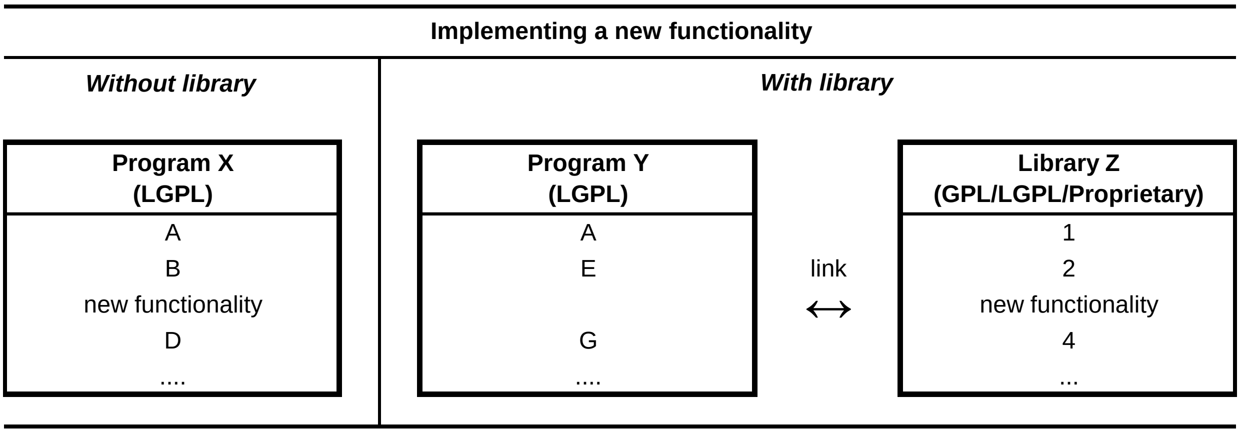


Figure 23: Adding a new functionality to GNU LGPL'ed software

The reason why licenses that allow dynamical linking with non-free modules such as libraries are only weakly copylefted is that it opens a possibility to not putting modifications under the same license. While changes in the weakly copylefted software itself must be put under the same license as well, this holds not true for the software that links to it. A software developer who does not want to put modified source code of weakly copylefted software under the same license may do so by linking to a non-free library and implement the functionality there. The modification can then be putted into a proprietary library or module and link to it. However, this is certainly not the intention of weak copyleft licenses but a corresponding abuse is legally possible. This backdoor in weakly copylefted licenses does not exist in strongly copylefted licenses.

Weakly Copylefted: The MPL Style

St. Laurent (2004: 63) concludes that the Mozilla Public License (MPL) constitutes a hybrid of the ideas of the GNU GPL and the BSD licenses. The MPL demands like the GNU GPL that changes in the source code are also released under the MPL. However, unlike strong copyleft licenses such as the GNU GPL source code under a MPL license may be combined with proprietary source code by dynamically linking to them. If a developer writes entirely separate software that merely links or calls to the library the weak copyleft license does not affect this separate software. However, at its emergence the MPL brought up also a new key issue not addressed by the BSD or GNU GPL licenses (Behlendorf 1999: 166).The MPL has several directions that protect the project as a whole as well as the individual developers against patent contestation. The license forces companies or individuals who contribute to a FOSS project to release any claims related to patent rights as far as the source code in question is concerned.[88]

[87] However, there is a special case. According to section three of the GNU LGPL software can be relicensed at any time under the GNU GPL (FSF 2005).

[88] This patent clause is subject to discussion because it may have far-reaching effects for firms holding patents. However, the reason why the topic of patents is addressed is to prevent somebody to contribute source code on which he has a

Weakly Copylefted: Other licenses

Another known weakly copylefted license is the GNU LGPL. Because it allows linking to proprietary software it is similar to the MPL. However, it does not contain any clauses that deal with patents. Other but less known examples of weakly copylefted licenses are the Sun Public License or the Nokia Open Source License.

Interestingly, Charles Hannum, cofounder of the NetBSD project, admits having had a very strong distaste for the copyleft clause in the GNU GPL license (Biancuzzi 2006). However, in retrospect he considers it as a problem of BSD style licenses that third parties do not have contribute back their changes.[89] The changes are then not only lost to the original project but as a result different and incompatible systems are produced. In consequence, Hannum stated that he probably would choose the GNU LGPL as a license if he started anew.

Strongly Copylefted: The GNU GPL Style

The vast amount of FOSS projects are licensed under the GNU GPL (see also chapter 2.5.2). Its copyleft mechanism mandates that modifications and derivatives have to be released under the GNU GPL as well.[90] This ensures that source code remains free and that there is no chance of forking into a commercial version (Behlendorf 1999: 167).[91] The copyleft mechanism is governed by the famous section 2b of the GNU GPL license (FSF 2006c):

> *2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:*
>
> > *a) …*
> >
> > *b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.*

It should be noted that the copyleft clause refers to the distribution and publication of source code. There is no legal need to reveal modifications of modified source code to third persons as long as the software is neither published nor distributed.[92] Strictly speaking, one can therefore not say that the GNU GPL forces to give modifications back to the community. However, in practice this is probably mostly the case because most software is somehow distributed or published.

---

patent, making the source code available to everybody, and then come back and charge for patent infringement (Mozilla Foundation 2007).

[89] In fact, the GNU GPL does legally not mandate that changes must be contributed back. However, *if* the changes are distributed they must be accompanied by the source code that practically leads to the discussed effect.

[90] What is a derivative work is not trivial to define and depends on the legal system. For a discussion of the topic see Välimäki (2005: 124-138). A controversially discussed case is binary-only drivers in the Linux kernel (see also chapter 2.6.1.1).

[91] The copyright holder is in fact able to license his source code under any possible license. However, this does not affect the terms of use of the source code that was putted under the GNU GPL.

[92] Additionally, running software licensed under the GNU GPL does not require accepting it. The section 0 of the GNU GPL makes clear that “activities other than copying, distribution and modification are not covered by this License; they are outside its scope” and that “the act of running the Program is not restricted” (FSF 2006c).

2.5.2 Distribution of licenses at SourceForge

SourceForge is a centralized hosting platform for FOSS projects. With more than 120’000 projects and 1’300’000 registered users it is the biggest FOSS hosting platform (SourceForge 2006a).[93] Other hosting platforms such as berliOS, GNU Savannah (only free software), Mozdev.org (only Mozilla applications and extensions), Sarovar (focus on India), and Tigris.org (only collaborative software development) are smaller and mostly more focused. Most of the large projects such as the Linux Kernel are not hosted on one of the mentioned hosting platforms. They have set up their own infrastructure.

Despite the fact that very large projects are hardly found on SourceForge it is a good reference point for research because of its size. In the following, the distribution of licenses at SourceForge will be discussed. In Figure 24 the licenses of the projects hosted at SourceForge are classified according to the lists and comments on licenses provided by ifrOSS[94] (2006), the FSF (2006e), and the OSI (2006a).

Licenses of SourceForge projects 2004 [a/b]

Classification	Style	License count	Project count	License count	Project count
Strong copyleft	GPL style	1	37'485	6	38'126
	Other	5	641		
Weak copyleft	MPL style	6	939	10	6'695
	Other	4	5'756		
Non copyleft	BSD style	14	6'052	24	6'778
	Other[c]	10	726		
FOSS[d]				6	14
Restricted[e/f]				3	1'184
Privileged[g]				2	1
Public Domain				1	1'524
Other / Proprietary[h/i]				n/a	975
All licenses (N)				52	55'297

[a] The data was collected at the 17th of May 2004.
[b] At the time the data was collected more projects were registered. However, not all projects declare their license.
[c] For some licenses it could not be determined whether they are in the BSD style. Such licenses are categorized as "Other".
[d] These licenses could neither be classified as copyleft nor as non-copyleft. This is due to their marginality which makes it difficult to find a reliable license review.
[e] Restricted licenses limit how modifications of software can be distributed. This category does not include copyleft licenses.
[f] Almost all projects in this category adopted the Artisitic License. The license versions prior to 2.0 are subject to discussion whether they qualify as free. Because there was no information about the license version the qualification of this license remains vague.
[g] Licenses with privileges grant the charactersitical rights of FOSS with special exceptions for a licensor.
[h] Apart the number of projects there is no further information about this category. Hence, the number of licenses can not be computed.
[i] There is disagreement between the Free Software Foundation and the Open Source Initiative whether the Apple Public Source License prior to version 2.0 is a FOSS license. Because there was on SourceForge no information concerning the version of the Apple Public Source License it is not sure whether the license qualifies as a FOSS license. In consequence, these projects were classified as "Other / Proprietary".

Figure 24: Licenses of SourceForge projects 2004[95]

[93] At June 15, 2006 122’520 projects and 1’333’382 users were registered at SourceForge.
[94] The ifrOSS is a German private institution dedicated to legal questions related to FOSS.
[95] Raw data from SourceForge (2004b).

The data presented in Figure 24 is consolidated in Figure 25 according to the typology discussed in chapter 2.5.1.2. Compared to the distribution of licenses at Freshmeat, another hosting platform, there are some similarities and differences (Freshmeat 2005). The share of strongly copylefted projects at Freshmeat is 68.06% and therefore almost identical to the corresponding share at SourceForge (68.95%). The share of weakly copylefted software at Freshmeat is with 6.47% considerably lower than at SourceForge (12.11%).

<table>
<tr><th colspan="5">Classified FOSS licenses at SourceForge 2004[a/b]</th></tr>
<tr><td>Strongly copylefted</td><td>68.95%</td><td>73.89%</td><td rowspan="2">86.86%</td><td rowspan="3">93.31%</td></tr>
<tr><td>Weakly copylefted</td><td>12.11%</td><td>12.98%</td></tr>
<tr><td>Not copylefted</td><td>12.26%</td><td>13.14%</td><td>13.14%</td></tr>
<tr><td>Other[c]</td><td>6.69%</td><td>-</td><td>-</td><td>6.69%</td></tr>
</table>

[a] The data was collected at the 17th of May 2004.
[b] N= 55'297
[c] "Other" consists the categories "FOSS", "Restricted", "Privileged", "Public Domain", and "Other / Proprietary".

Figure 25: Classified FOSS licenses 2004[96]

[96] Raw data from SourceForge (2004b).

In the year 2006 the following licenses were found at SourceForge (Figure 26):

Licenses of SourceForge projects 2006[a/b]							
Classification	Style	License count	Project count	License count	Project count	Change of licenses since May 04	Change of projects since May 04
Strong copyleft	GPL style	1	53'877	7	55'793	+1 (+17%)	+17'667 (+46%)
	Other	6	1'916				
Weak copyleft	MPL style	8	1'735	12	11'412	+2 (+20%)	+4'717 (+70%)
	Other	4	9'677				
Non copyleft	BSD style	14	9'885	24	12'483	+0 (+0%)	+5'705 (+84%)
	Other[c]	10	2'598				
FOSS[d]				9	247	+3 (+50%)	+233 (+1'664%)
Restricted[e/f]				3	1'420	+0 (+0%)	+236 (+20%)
Privileged[g]				2	6	+0 (+0%)	+5 (+500%)
Public Domain				1	2'459	+0 (+0%)	+935 (+61%)
Other / Proprietary[h/i]				n/a	1'457	n/a	+482 (+49%)
All licenses (N)				58	85'277	+6 (+12%)	+29'980 (+54%)

[a] The data was collected at the 15th of June 2006.
[b] At the time the data was collected more projects were registered. However, not all projects declare their license.
[c] For some licenses it could not be determined whether they are in the BSD style. Such licenses are categorized as "Other".
[d] These licenses could neither be classified as copyleft nor as non-copyleft. This is due to their marginality which makes it difficult to find a reliable license review.
[e] Restricted licenses limit how modifications of software can be distributed. This category does not include copyleft licenses.
[f] Almost all projects in this category adopted the Artistic License. The license versions prior to 2.0 are subject to discussion whether they qualify as free licenses. Because there was no information abouth the license version qualification of this license remains vague.
[g] Licenses with privileges grant the charactersitical rights of FOSS with special exceptions for a licensor.
[h] Apart the number of projects there is no further information about this category. Hence, the number of licenses can not be computed.
[i] There is a disagreement between the Free Software Foundation and the Open Source Initiative whether the Apple Public Source License prior to version 2.0 is a FOSS license. Because there was on SourceForge no information concerning the version of the Apple Public Source License it is not sure whether the license qualifies as a FOSS license. In consequence, these projects were classified as "Other / Proprietary".

Figure 26: Licenses of SourceForge projects 2006[97]

In a condensed form the following licenses were found in 2006 at SourceForge (see Figure 27):

Classified FOSS licenses at SourceForge 2006[a/b]				
Strongly copylefted	65.43%	70.01%	84.34%	93.45%
Weakly copylefted	13.38%	14.32%		
Not copylefted	14.64%	15.66%	15.66%	
Other[c]	6.55%	-	-	6.55%

[a] The data was collected at the 15th of June 2006.
[b] N= 85'277
[c] "Other" consists the categories "FOSS", "Restricted", "Privileged", "Public Domain", and "Other / Proprietary".

Figure 27: Classified FOSS licenses at SourceForge[98]

[97] Raw data from SourceForge (2006b).

In Figure 28 the change from 2004 to 2006 in the number of projects under a certain license is presented. As can be easily seen the *absolute* number of projects has grown for all categories. More telling is the *relative* change in the number of projects under a certain license. Figure 29 presents the ten largest losers in *relative* share across all licenses and the biggest winners.

<table>
<tr><th colspan="5">Change of classified FOSS licenses at SourceForge from 2004 to 2006[a/b]</th></tr>
<tr><td>Strongly copylefted</td><td>+17'667 (+46%)</td><td>+17'667 (+46%)</td><td rowspan="2">+22'384 (+50%)</td><td rowspan="3">+28'089 (+54%)</td></tr>
<tr><td>Weakly copylefted</td><td>+4'717 (+70%)</td><td>+4'717 (+70%)</td></tr>
<tr><td>Not copylefted</td><td>+5'705 (+84%)</td><td>+5'705 (+84%)</td><td>+5'705 (+84%)</td></tr>
<tr><td>Other[c]</td><td>+1'891 (+51%)</td><td>-</td><td>-</td><td>+1'891 (+51%)</td></tr>
</table>

[a] The data was collected at the 17th of May 2004 and at the 15th of June 2006.
[b] N 2004= 55297; N 2006= 85'277
[c] "Other" consists the categories "FOSS", "Restricted", "Privileged", "Public Domain", and "Other / Proprietary".

Figure 28: Change the number of classified licenses from 2004 to 2006[99]

Change in the share of all licenses at SourceForge from 2004 to 2006					
	The biggest losers (change in %)			The biggest winners (change in %)	
1	GNU GPL	-4.191	***1***	Apache License 2.0	1.287
2	Artistic License	-0.495	***2***	GNU LGPL	0.977
3	Apache Software License	-0.332	***3***	Academic Free License	0.597
4	Mozilla Public License 1.0	-0.169	***4***	Common Public License	0.430
5	Python License (CNRI Python License)	-0.119	***5***	Open Software License	0.396
6	Qt Public License	-0.105	***6***	PHP License	0.352
7	Other/Proprietary License	-0.045	***7***	BSD License	0.341
8	Sun Industry Standards Source License	-0.026	***8***	Mozilla Public License 1.1	0.300
9	Jabber Open Source License	-0.014	***9***	MIT License	0.280
10	Eiffel Forum License	-0.010	***10***	Public Domain	0.147

Figure 29: Change in the share of licenses at SourceForge from 2004 to 2006[100]

Some changes in the share of all licenses are easy to explain respectively not important for this study. The decline of the Mozilla Public License 1.0 and the Apache Software License are probably just a upgrading of projects to improved license version. This partly explains also the rise of the Apache License 2.0 and the Mozilla Public License 1.1. The decline of the original Artistic license may be due to its vague license terms that caused problems and legal uncertainty (FSF 2006e). Trolltech relicensed its Qt library in 2000 under a double Qt Public License and GNU GPL license.[101] This enabled FOSS projects to switch from the Qt Public

[98] Raw data from SourceForge (2006b).
[99] Raw data from SourceForge (2004b; 2006b).
[100] Raw data from SourceForge (2004b; 2006b).
[101] Related to this relicensing was the "GPL versus Qt War" (Sweet 2001: 431-432). In response to the licensing policy of Trolltech the GNOME desktop environment project was started in 1997. This occurrence and the relicensing of the QT library were very important in FOSS licensing history.

License to the GNU GPL. It may be that not all projects switched instantly and the observed decline in the share of all licenses of the Qt Public License is just an aftermath of the relicensing.

However, there is an interesting change in relative share that cannot be explained here easily. There is a sharp decline in relative share of the GNU GPL. This trend can additionally be corroborated with data from the year 2001.[102] From 2001 to 2004 the BSD license together with the MIT license had a higher growth rate (+283% %) than the GNU GPL (+220%). In 2001 there were at SourceForge *ten* times more GNU GPL projects than BSD and MIT projects together, in 2004 *eight* times more, and in 2006 this proportion dropped to *seven*. Additionally, the Apache License and the GNU LGPL have risen in relative importance. As a working hypothesis it can therefore be concluded that the popularity of non-copylefted and weakly copylefted licenses grow faster than the popularity of strongly copylefted licenses. However, to corroborate this interesting and highly relevant hypothesis further research is needed (see chapter 12.4.3). It may be that the observation is just an artifact.

2.5.3 The Netscape - Mozilla - Firefox example

In the mid-nineties the Netscape Navigator was the most popular web browser. Alarmed by the enormous growth of the Internet Microsoft started to push its own web browser Internet Explorer. Very soon the Internet Explorer gained momentum and by the end of the nineties the Netscape Navigator lost its relevance. Some argue that one major reason for the success of the Internet Explorer was Microsoft's bundling of the web browser with the Windows explorer (Jackson 1999: 204). Microsoft also seemed to have refused OEM's request to ship a Windows version without the web browser and prevented its removal by OEM's. Howsoever, the Netscape was forced to react to the sharply declined market share. Netscape finally decided to put the source code under a FOSS license and to start the Mozilla project to benefit from the power of the FOSS community (see also chapter 2.2). This move was highly noticed because it was the first important commercial firm that relied on FOSS development. The FOSS community itself had also big expectations.

However, only a few FOSS developers joined the Mozilla project and in consequence mainly developers on Netscape's payroll contributed source code. There were different problems associated with the project such as low modularity of the source code and the charge of being bloatware.[103] A major problem was also the licensing scheme. Some parts of the source code were licensed under the Netscape Public License (NPL). This license had three major defects (Stallman 2005):

i. ***Not all users are equal:*** The terms of the NPL grants Netscape some special rights. It allows Netscape to put revised source code under a proprietary license again. Principally, this is not thought to be a bad thing because BSD style licenses allow such procedures as well. It was the asymmetry of rights that caused uncomfortable feelings. As Stallman argues the deeper message of the NPL denies the idea of cooperation among equals in the community.

ii. ***Not a true strong copyleft license:*** The NPL has the form of a copyleft license because modifications must be released under the NPL as well. However, the license terms do not include the special case of added subroutines if they are put in a separate file. In consequence, one may make changes to the

[102] The data from August 8, 2001 was collected by Feller and Fitzgerald (2002: 18 and 20). There were 11'723 projects under the GNU GPL and 1228 projects under a BSD or MIT license.

[103] Bloatware is software with low functionality relative to its use of diskspace and memory (Raymond 2003, bloatware).

source code proprietary by putting them into such subroutines. Again, not being a strong copyleft license is not a drawback because the BSD license and the GNU LGPL are neither. The problem is that the NPL looks like a real strong copyleft licenses without being one.

iii. ***Not compatible with the GNU GPL:*** A practical problem of the NPL is that it is not compatible with GNU GPL. This makes it impossible to combine source code licensed under the NPL and the GNU GPL. In practice this means that it gets difficult to reuse source code. This forces developers to write unnecessarily source code again and makes development inefficient.

As discussed above, the terms of the NPL license had practical problems and did as well not correspond to the norms of the FOSS community. Eric Raymond advised Netscape explicitly that the FOSS community reacts very sensitive to licensing issues (Young & Rohm 2000: 128-132). He noted that the future of the Mozilla project depends on the goodwill of the community. Raymond anticipated correctly that the community would not have join the project if they had thought that there are licensing problems.

Because of the resistance against the NPL the Mozilla project stared relicensing the source code in September 2001 under a MPL/GPL/LGPL triple license (Mozilla Foundation 2006a). Having solved the license problems the Mozilla project gained momentum. The Firefox web browser that is based on the Mozilla layout engine Gecko is now a success story and heavily used. From the Netscape - Mozilla - Firefox story a very clear conclusion can be drawn. Proper licensing as well as adherence to the FOSS norms is very important for everybody who wants to cooperate with the community.

2.6 Protecting mechanisms

Everything valuable is probably in danger of being appropriated legally or illegally. Therefore, it does not surprise that there are threats to FOSS. The different threats are presented first and afterwards the corresponding solutions are discussed.

2.6.1 Threats to FOSS development

Below three different but to some extent overlapping threats to FOSS are discussed. These threats are based on i) commercial firms who do not adhere to community norms and license terms, ii) on developers who do not adhere to community norms and license terms, and iii) on malware and reputation losses by inappropriate behavior of certain actors.

Although the list of threats presented below is not exhaustive it is thought that it covers the most important aspects. It must be stated that not everybody agrees that the presented threats are important or that they exist.

2.6.1.1 Commercial firms

While the FOSS community seeks to protect the commons single commercial firms seek to prevent the revealing of source code to the FOSS environment (Dahlander & Magnusson 2005; Henkel 2006). This may counteract the goals of FOSS development.

Take but not give back: Non-copyleft licenses

A characteristic of non-copyleft licenses is that modified source code can be taken private.[104] As already discussed in chapter 2.5.1.2 this may lead to the problem that commercial firms profit from FOSS projects by incorporating source code into their own software but do not give the modified source code back. According to the terms of non-copyleft licenses it is completely legal to appropriate source code and keep improvements private. An example of a project that probably suffered from the described problem is the Apache HTTP server (Behlendorf 1999: 165).

Take but not give back: Copyleft licenses

Copyleft licenses are constructed so that the above-mentioned problems should be eliminated. However, practically the problems remain. In a study based on qualitative and quantitative data Henkel (2006) researched how commercial firms in the embedded[105] Linux market refrain from making source code available. Because Linux is licensed under the GNU GPL derivative source code must be revealed to customers. This necessity seems to be quite harsh in preventing to not reveal source code. However, there are legal, probably illegal, and illegal mechanisms to refrain from doing so (Henkel 2006: 959):

i. ***Revealing only to customers:*** The GNU GPL does not require making derived work *public*. It requires that the seller respectively the distributor of a derived work must make the source code available *to customers* respectively *recipients* of the software. If there are few recipients of the software and they do not want to get the source code it can be kept secretly.

ii. ***Revealing only on request:*** If the device is sold to the mass market the distributor can delay the diffusion of source code by not actively distributing it and only providing it on demand. The GNU GPL allows to copy and distribute the software if it is accompanied "with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code" (FSF 2006c, section 3b).

iii. ***Delay revealing:*** One can delay the revealing of source code on purpose.

iv. ***Loadable binary modules:*** Some firms make drivers available only as a loadable binary module and not as source code.[106] Although the license terms of the GNU GPL related to copyleft and derived work seems to be clear that this is not legal there is a discussion on this topic in the Linux kernel project. It is questioned whether loadable modules can be considered as derived work under copyright law. A problem is that it is not easy to find out whether a work is derived or not (Torvalds 2003).

While the two mechanisms discussed first are probably not appreciated they are completely legal. The third and the especially the fourth mechanisms are highly disputed or even illegal.

---

[104] A fraction of the FOSS community argues that software is only free if modified source code can be taken private (see chapter 2.3.2).

[105] Embedded systems are designed for special purposes and predefined tasks. Such computer systems are adapted to special requirements and only adapted software can be run. Examples are automatic teller machines, mobile phones, copiers, air conditioner, elevators, or microwave ovens.

[106] For a discussion of the topic see the whole threads in the Linux Kernel Mailing List following the emails of van de Ven (2001), Hellwig (2002), and Bennet (2003).

The most disputed mean for not revealing source code, only providing binary modules, is also the most frequent (see Figure 30). More than half of the firms use binary only modules always, often, or at least sometimes. Revealing code only after a delay is the least popular mean to not reveal source code.

Frequency of means for not revealing source code							
Means of protection	**Always**	**Often**	**Sometimes**	**Rarely**	**Never**	**N**	**Missing**
Revealing only to customers	7.0%	14.0%	25.6%	11.6%	41.9%	43	17
Revealing only on request of device buyers	5.8%	22.5%	16.7%	16.7%	38.4%	138	59
Revealing only after delay	2.8%	11.4%	21.3%	21.3%	44.7%	141	56
Loadable binary modules	16.1%	19.5%	17.5%	17.5%	36.2%	149	48

Figure 30: Means to prevent the revealing of source code[107]

While only 49.3% of commercial firms actively make source code public 92.0% of non-commercial organization (e.g. universities) and individuals (e.g. hobbyists) do so (Henkel 2006: 960). However, it seems that commercial firms change their attitude towards revealing source code. Almost 50% of commercial firms state that they reveal right now (in late 2003 / early 2004) somewhat or much more source code than they did in the year 2000. Only 10% of commercial firms stated that they reveal less while 41% did not see a change.

Software patents

Software patents are probably one of the most heated areas of debate in patenting (OECD 2004: 24) and they are considered within the FOSS community as a very big threat (see e.g. Stallman 2006).[108/109] The share of patents related to information and communication technology increased from the late 1990s to the beginning of the 2000s rapidly (OECD 2006: 18-19). In countries such as the Netherlands or Finland ICT-patents already accounts for more than half of all filed patents. In the computer hardware industry this is probably due to i) the nature of the underlying technology that fosters incremental innovation, ii) the rise of defensive patenting, and iii) the ease of obtaining patents (FTC 2003: 34-35, chapter 33).

It is argued that patents are suitable in providing incentives for innovation with high development cost, for innovation that is easily imitated, and for innovation for which a patent can be clearly defined such as in pharmaceuticals (Hall 2003: 16).[110] However, if innovations are incremental and when different innovations must be combined such as in software development it is questionable whether the benefits of a patent system outweigh the costs. Luethi (2005: 6-9) argues that software patents raise the cost of production considerably:

[107] Henkel (2006: 959).

[108] For a short discussion of a definition for software patents see Bessen and Hunt (2004: 7-8).

[109] Impressive examples of granted European patents related to trivial functions in web shops are provided by the FFII (FFII 2007b).

[110] Justice Antonin Scalia of the U.S. Supreme Court described the current standard for defining an invention as obvious with the term “gobbledygook” (Samardzija 2007: 190).

- ***Patent costs:*** The patent attorney of SAP Harald Hagedorn (2003) estimates that the average cost for a software patent application is between 20’000 and 50’000 euros. Hall et al. (2003: 23) estimate for the United States 34’000$ for the application of a patent. If a reexamination is started which may be initiated by any other party during the life of the patent there are additional estimated costs between 12’520$ and 102’520$.

- ***Litigating costs:*** The cost of a patent litigation is between $1 and $3 million (Somaya 2003: 17).[111]

- ***Patent thicket:*** Software can contain million of lines of code and include potentially an uncountable number of patentable inventions (FTC 2003: 26, chapter 22). As Peter Detkin from Intel states in the FTC hearings on the balance of competition, patent law, and policy there are more than 90’000 patents related to microprocessors held by more than 10’000 parties (FTC 2003: 34, chapter 33). There are approximately 420’000 semiconductor and system patens held by more than 40’000 parties. As Detkin observes further the patens unavoidable overlap each other. Another panelist of the FTC hearings found that for his software product 120 patents overlap each other (FTC 2003: 52, chapter 53). This produces a patent thicket that generally occurs if innovation is incremental such as in the software industry. It is in consequence not astonishingly that mutual hold up with patents is especially high in the computer industry (Somaya 2003: 34). The patent counsel of IBM states that there are two different values based on patents: *Firstly,* trough license fees and *secondly* through licensing negotiations that give access to patents from other firms (Grindley & Teece 1997: 15). He estimates that the value generated by access to other patents is ten times larger that the value of license fees.

FOSS developers as well as small and medium enterprises can obviously hardly afford the costs discussed above (Luethi 2005: 9). Therefore, these actors are aggrieved by the existence of software patents.

Some authors question the claimed benefits of patents. Based on empirical data Bessen and Hunt (2004: 41) argue that granting software patents in the U.S. during the 1990s may have resulted in a decrease in innovation. Software patents substitute for R&D and are associated with lower R&D intensity (Bessen & Hunt 2004: 30). Bessen and Hunt (2004: 32-33) estimate that by the end of the 1990s R&D would have been about 10% higher without this substitution effect. Similarly, the panelists of the mentioned FTC hearings were critical whether patents foster innovation for computer hardware (FTC 2003: 32, chapter 33):

- Robert Barr (Cisco Systems): “the company had grown to over a billion dollars in annual revenue. This growth was obviously not fueled by patents, it was fueled by competition and by open, nonproprietary interfaces.”

- Desi Rhoden (Advanced Memory International): “competition is what drives . . . innovation; patents have almost nothing to do with innovation.”

- Gary Zanfagna (Honeywell International): “innovation is driven by competition in all of our markets.”

Similarly, Yar Chaikovsky (Zaplet Inc), Robert Kohn (Borland Software Inc), Bradford Friedman (Cadence Design System), and Mary Musacchia (SAS Institute) argued that for software competition drives innovation and that the patent system does not encourage innovation (FTC 2003: 46, chapter 43). An example is Yahoo that reached a $120 billion market capitalization with only three issued patents (FTC 2003: 49, chapter 43).

[111] Hall et al. (2003) provide inconsistent data. At page 23 the estimated litigation costs are between $0.5 and $3 million while at page 7 the same costs are between $1 and $3 million.

2.6.1.2 Malware and reputation losses

The freedom to run, study, adapt, improve, and redistribute or release source code is by definition associated with FOSS. This includes also the right to change the source code so that the modified software is *inferior*. The addition of a new feature may result in perceptually software crashes including loss of data. Even worse the “improvements” may include unwanted or harmful functionalities such as seen in malware, spyware, or adware. A FOSS browser may be forked by a commercial firm. An added feature may display pornographic ad and direct users to corresponding websites. Depending on the specific characteristics of the software and the terms of the license the installation of such software may be legal because users have given their consent to installation by accepting the end-user license agreement (Kunkel 2002).

Certainly, the problems of bad software, malware, spyware, or adware do also exist in the case of closed source or proprietary software. However, the probability that modified software is associated with the original project or organization independent of negligence is probably more likely for FOSS projects than for commercial firms. Reputed commercial firms that own valuable software will hardly give consent to forks that include malware, spyware, or adware.

In either case the modified software may hurt the reputation of the original FOSS project. This is all the more a problem because “… when it comes to intangible products like software, trust is all consumers have to decide on” (Mozilla Foundation 2006b).

2.6.1.3 Developers who do not adhere to community norms

Individual developers who do not follow the terms of a license commit the same wrongdoing as firms. However, it seems to be quite clear that in practice individual developers and firms are somehow different. *Firstly*, most individuals have not the knowledge to fully understand the terms of licenses. They are not able to hire lawyers to research the terms of a nameless license. *Secondly*, FOSS applications produced by individuals are probably less likely distributed widely. Modified FOSS applications are then probably only for personal use. This has legal as well as practical implications. The GNU GPL does only require putting modifications under the same license if the software is distributed. It is also practically hardly controllable what a developer has modified if he does not distribute the software. *Thirdly*, the amount of individual developers is quite high and to ask everybody to comply with license terms or even to litigate them might be laborious.

2.6.2 Solutions to protect FOSS development

In a qualitative study using ethnographic methods O’Mahony (2003) discusses the question how FOSS projects managed to protect themselves from threats. He identifies seven tactics (O'Mahony 2003: 1183):

i. Adoption of a license that restrict appropriation

ii. Encouragement of compliance with licensing terms through normative and legal sanctions

iii. Incorporation to hold assets and protect individual contributors from liability

iv. Transfer of individual property rights to collectively managed non-profit corporations

v. Trademark the brands and logos designed to represent their work

vi. Assignment of trademarks to a foundation

vii. Active protection of the project's brand

One may add four additional tactics:

viii. Consider licenses as constitutions

ix. Enforcement of license terms by non project specific non-profit organizations

x. Fight against software patents

xi. Patent pools

This tactics can be summed up in the following six broader categories that will be discussed in turn below:

i. Use of copyleft licenses

ii. Project specific non-profit organizations to enforce licenses

iii. Non-project specific non-profit organizations to enforce licenses

iv. Patent pools and fight against software patents

v. Adoption of trademarks

vi. Consider licenses as constitutions

2.6.2.1 Use of copyleft licenses

FOSS licenses grant permission to do things that is otherwise prohibited because of copyright law. However, this permission is only granted if one follows the terms and conditions of the license. By choosing a proper license one may restrict undesired effects such as for example appropriation.[112]

As already discussed in chapter 2.5.1.2 there are special types of FOSS licenses that are called strong copyleft licenses. The characteristic of copyleft licenses is that it ensures by its term that the granted rights also apply to derived and modified versions. If source code is licensed by a copyleft license there is no legal way for putting it under a proprietary license.[113] As Moglen (1999) puts it: "It {copyleft} creates a commons, to which anyone may add but from which no one may subtract". There is the permission to run, copy, modify, and distribute the software but not to add any additional restrictions on the software (Stallman 2001a).

As already mentioned examples of copyleft licenses are the GNU GPL, the Common Public License, the IBM Public License, the Jabber Open Source License, the Eclipse Public License, and the NetHack General Public License. However, of all projects at the FOSS hosting platform SourceForge that adopted a strong copyleft license 97% chose the GNU GPL (see chapter 2.5.2).

It can be summarized that copyleft licenses are very good and easy mechanisms to prevent that commercial firms take but not give back.

[112] For software in the public domain copyright law does not apply and one may therefore not prevent such undesired effects.

[113] However, although it is not possible to relicense the source code under a proprietary license the copyright holder may put it under another license that could be proprietary. In practice this works only for small projects with few copyright holders or commercial FOSS projects that do not accept contributions form the outside. In the Linux kernel there are a multitudinous number of developers who contributed. To put Linux under a proprietary license all these developers must agree which is hardly likely.

2.6.2.2 Project specific non-profit organizations to enforce licenses

Some FOSS projects such as Apache, Perl, FreeBSD, GNOME, KDE, Python, and Jabber have established foundations. The purposes of these non-profit organizations are manifold. Frequently, the reason for doing so is to provide infrastructure, press coverage, and to collect money. Furthermore, FOSS foundations own the trademarks of names, phrases, logos, or symbols and the copyright of the source code. This does not only protect individual contributors from liability but also facilitates the enforcement of legal sanctions. The problem of a community is that it is not a legal actor (O'Mahony 2005: 395).

However, the problem with foundations is that individuals contributing to a community project want to recognize each other as individuals and retain their individual autonomy (O'Mahony 2005: 410-411). It may be a potential threat to the FOSS community model that institutions structure the development in a way not appreciated by FOSS programmers. The unbureaucratic way of producing software may be altered in order to comply with institutional requirements. However, so far no such problem seems to have emerged.

2.6.2.3 Non-project specific non-profit organizations to enforce licenses

There are mainly two non-project specific organizations that are concerned with GNU GPL enforcement: gpl-violations.org managed by Harald Welte and the compliance lab of the FSF (Linux Weekly News 2006). The organization gpl-violations.org is based on the German legal system. It enforces the GNU GPL in Germany but also in the rest of the world such as in Taiwan or Korea. The FSF is based on the legal system of the United States. It can only enforce violations if the copyright has been transferred to the FSF.

Harald Welte (2006) states that the rate of new GNU GPL violations is alarming. According to his observation there is no Linux-based NAS (Network-attached storage) product that is compliant to GNU GPL license terms the first time it is shipped. The problem of GNU GPL violations seems to be especially dramatic in the embedded market (see also chapter 2.6.1.1). However, although the absolute rate of GNU GPL violations is increasing, they are relatively decreasing. This is due to the massive growth of the Linux embedded market.

There are not many examples of court decisions related to the GNU GPL. An exception was the lawsuit between Sitecom and Harald Welte concerning the FOSS project Netfilter/iptables. According to the Landgericht München I, a district court in Germany, Sitecom violated the GNU GPL license terms (Landgericht München I 2004). More important the court concluded that it has no doubts about the permissibility of the conditions (Landgericht München I 2004: 18).

Although there are not many court decisions concerning the GNU GPL there are many successful enforcements in which subjects who violated the terms agreed to comply in out of court settlements (Linux Weekly News 2006; Moglen 2006). According to Harald Welte there were in the 12 months prior to June 2006 158 reports of GNU GPL violations of which about 100 were real violations (Linux Weekly News 2006). About 40 of these violations were resolved while the others were not yet resolved. Since the start of the gpl-violations.org project in January 2004 there has been no single unsuccessful enforcement of a GNU GPL violation. One factor for this success is that the GNU GPL is a license and therefore subject to copyright law and not a contract subject to contract law (Groklaw 2003).[114] While a license is a unilateral permission to someone else's property, a contract is an exchange of obligations. The GNU GPL for example is a unilateral permission for which no reciprocal obligation is required. In copyright law nobody is allowed from copying, distributing, and making

[114] The correctness of this statement does not hold true for all legal systems.

derivative works if he is not granted permission. This is a strong protection mechanism. If one did not accept the license term or a court were to judge a condition invalid all rights related to the license would be withdrawn.

2.6.2.4 Patent pools and fight against the possibility to patent software

The FOSS community reacts to the threat of software patents with two different approaches. Within the community the most appreciated solution to the problem is to prevent respectively abandon the possibility to patent software (see for example FFII 2007a; FSF Europe 2006).

Another but disputed (Broersma 2005) solution is the creation of patent pools to protect FOSS projects and developers. In patent pools actors agree to cross-license patents.[115] Basically, there is a reciprocal sharing of patents between actors without the exchange of a fee. Shapiro (2001: 1) argues that cross-licensing and patent pools are especially important in industries such as computing in which formal standard setting is vital for bringing new technologies to market. If there is a patent thicket in which all actors have blocking patents this may be the only possibility to produce goods.

An example of a patent pool is the Patent Commons Project launched by the Open Source Development Labs in November 2005 (OSDL 2005).[116/117] The commons "contains pledges, promises, covenants and other agreements … that have been made in support of open source software and standards" (Patent Commons Project 2007b). The related "commitments are enforceable promises in which patent holders agree they will not, under certain terms and conditions, assert patent rights against third parties who are engaging in activities that might otherwise give rise to a claim of patent infringement" (Patent Commons Project 2007a).

2.6.2.5 Adoption of trademarks

By the adoption of trademarks on logos, names, sounds, and the like FOSS projects can protect themselves from being muddled up with malware, spyware, adware, or dysfunctional modifications. Thereby FOSS projects may protect their reputation. The trademark policy of FOSS projects may require that some rules are followed if the logo or the name of the project is used. For example one may demand that the software can only be distributed under the project's name if the official binaries are unchanged.

The Mozilla Foundation (2006b) that develops the Firefox web browser has quite a strict trademark policy:

> *Mozilla's need to ensure that its trademarks remain reliable indicators of quality and security. … when it comes to intangible products like software, trust is all consumers have to decide on.*

That the Mozilla Foundation does enforce its trademark policy very strictly can be recognized by the debate between the Mozilla Foundation and Debian (see Debian Project 2007). Debian, a big and very much respected Linux community-distribution, included the Firefox web browser in their Linux distribution. The Debian guidelines (Debian Project 2004) do not tolerate non-free logos or icons. That is why Debian refused to use the

[115] For Kato (2004: 256) cross licensing is different from patent pools insofar as the former involves only two parties.

[116] The Open Source Development Labs and the Free Standards Group have merged to the Linux Foundation in January 2007.

[117] Another example is the Open Invention Network (2006). For a short comparison of the two projects see Jones (2005b).

Firefox logos.[118] They incorporated only the name Firefox but not the Firefox logos and icons in their distribution. However, according to the trademark policy of the Mozilla Foundation the name cannot be used without the Firefox logo (Mozilla Foundation 2005).[119] Due to the enforcement of the strict trademark policy by Mozilla Foundation Debian had to change the name of the Firefox web browser in their distribution to Iceweasel.

2.6.2.6 Consider licenses as constitutions

As Luhmann (1987) argues there are different functions of law. One function is reducing complexity. Law informs us how to act (134). The influence of the copyleft terms can then not be solely reduced to its legal implications. It also serves as a normative guidance that informs developers what is beneficial to the cause and what is not. It can be assumed that in low-rated projects developers could get away with offenses of the GNU GPL but it is not done because it is thought that it should not be done. Reciprocity in FOSS development is then not only fostered by the threat of legal sanctions but also by the normative guidance written down in the legal terms. Basically, the GNU GPL tells developers that sharing and revealing of source code is good and beneficial. As Moody (1997) argues the GNU GPL provides:

> *… a written constitution for the new online tribe of Linux hackers. The license said it was OK to build on, or incorporate wholesale, other people's code - just as Linux did - and even to make money doing so (hackers have to eat, after all). But you couldn't transgress the hacker's fundamental law of software: source code must be freely available for further hacking.*

In a very interesting approach Weber (2004: 179) argues similar to Moody that licenses are constitutions. He notes that in the absence of hierarchical authority the license becomes the core statement of the social structure that defines the community of FOSS developers. The core message of a FOSS license is a statement of and for the developers. The related principles for the Debian community are (Weber 2004: 180):

- The freedom to give away or sell the software, as well as to modify it, through access to source code.
- Nondiscrimination against persons, groups of people, or particular fields of endeavor.
- Pragmatism respectively that there should be not restriction for using proprietary software.
- Meritocracy that includes an equal opportunity to succeed or fail based on how good one is as a coder.

Weber (2004: 181-185) discusses further the constitutional meaning of licenses such as the BSD and the GNU GPL. For community developers working on a FOSS project means implicitly accepting the social contract stated by license terms (Weber 2004: 182). This seems to be of a similar importance as the legal meaning of licenses. The Debian project has explicitly ratified the Debian Social Contract as an addition to FOSS licenses (Debian Project 2004). This social contract states a set of commitments such as to “give back to the free software community”, to “remain 100% free”, or to “not hide problems”.

Similarly, Välimäki (2005: 9) notes that one may argue that community norms of FOSS projects reflect the corresponding terms of a license. For Osterloh and Rota (2007: 167) FOSS licenses are an important institutional innovation that restrict appropriation and that are a codified expression of a norm of reciprocity.

[118] The only exception to use non-free material is related to FOSS project names (Debian Project 2004). That is why Debian used the Firefox name but not the logos.

[119] However, the Firefox logo can be used without the Firefox name (Mozilla Foundation 2005).

To ensure that developers adhere to community norms the choice of an appropriate license seems to be very important. By choosing a known license such as the GNU GPL or a BSD license one releases individual developers to analyze the terms of the license. Within the corresponding communities the knowledge is already available.

Franck and Jungwirth (2003b: 403) argue that the GNU GPL respectively its copyleft term is an institutional mechanism with two important characteristics. *Firstly*, software under a GNU GPL license sends a credible signal to rent-seekers "that nobody will be able to cut them off from the stream of attention generated by their contributions" (Franck & Jungwirth 2003b: 409). *Secondly*, the copyleft term acts as a non-distribution constraint for developers with a pro-social motivation (Franck & Jungwirth 2003b: 411-413). Hansmann (1980) coined the term of non-distribution constraint that excludes the distribution of "net earnings, if any, to individuals who exercise control over it".[120] Project leaders cannot appropriate the software unfairly. Every distributed modification of software under the GNU GPL is available to all interested individuals. If a project member is not happy with the project leader he can start to fork of the project without loosing any source code.

An example that shows the importance of the constitutional meaning of FOSS licenses is the protest following the Microsoft-Novell patent agreement (see Groklaw 2006c). In this agreement Microsoft pledged to not litigate Novell Linux customers, openSUSE[121] contributors, and unsalaried FOSS developers. This agreement was highly criticized in the FOSS community. Jeremy Allison, a known hacker of the Samba project and employee of Novell, wrote to the management to warn them to sign the agreement (Groklaw 2006a):

> *… even if it does not violate the letter of the license, it violates the intent of the GPL license … whilst the Microsoft patent agreement is in place there is *nothing* we can do to fix community relations. And I really mean nothing.*

The management of Novell does not seem to have taken this concern seriously. In consequence, Allison left Novell and was instantly hired by Google. Additionally, Novell seem to have lost much credit in the FOSS community.

Licenses as expressions of norms are not only important for individuals but also for commercial firms. For Benkler (2002: 445) the capacity of a firm to show credibly to not appropriate a FOSS project unfairly is crucial for a business model based on a peer production model. A very credible commitment is to choose an appropriate license (see also Franck & Jungwirth 2003b: 409). As Benkler (2002: 379-380) argues the GNU GPL prevents defection in the development of the public good FOSS.

Law is not only result of social change but also induces social change itself (Luhmann 1987: 294). With the time going on legal terms may therefore have a very deep influence on cooperation and product development in the IT sector.

2.6.2.7 Threats to FOSS development and the corresponding solutions: Overview

Figure 31 provides a short overview over threats and possible solutions in FOSS development.

---

[120] For a discussion of the non-distribution constraint and intrinsic motivation see Valentinov (2007).

[121] OpenSUSE is the community version of Novell's Linux distribution.

	Threats		Solutions
Coming from commercial firms	Take but not give back: Non-copyleft licenses	►	Use copyleft licenses
	Take but not give back: Copyleft licenses	►	Project specific non-profit organizations
		►	Non project specific non-profit organizations
	Software patents	►	Fight against software patents
		►	Patent pools
	Malware and reputation losses	►	Adoption of trademarks
Coming from developers	Developers who do not adhere to community norms	►	Consider licenses as constitutions

Figure 31: Threats to FOSS development and the corresponding solutions

2.7 Forking

In FOSS development forking is a relatively rare but very important phenomenon. It happens if developers take a copy of source code from a FOSS project and start a new project independently. Two or more versions of a software package's source code are being developed in parallel (Raymond 2003, fork). Although the projects shared once a common code base they may get (but do not must) incompatible with development going on.

2.7.1 Forks versus project variants

Creating a variant of a project’s code respectively a development branch is not a fork unless there is an intent to create a competing project (Wheeler 2005). Furthermore, releasing variants for experimentation may be part of the regular FOSS development process. FOSS projects such as the Linux kernel intentionally have “fly-offs” (also called “bake-offs”) with different competing implementations. The results are discussed in evolutionary terms and the best result respectively the “winning mutation” is accepted by the project while the alternatives are abandoned “evolutionary dead ends”. It is therefore the *intent to replace* an original FOSS project itself that distinguishes a fork from a project variant.

2.7.2 Reasons to fork a project

There are several reasons to fork a project:

- ***Technological:*** There may be different incompatible possibilities for further development. This may for example include the question whether a software package should mainly run fast and avoid a software bloat or whether it should offer as many features as possible.
- ***Project management:*** Developers may not be happy with the project management. A project may be practically unmaintained or leaders accept changes too fast or too slow.

- ***Core values:*** Closely related to project management there may be disagreements about core values of the project. This may include for example disagreement about the license, decision making procedures, and the alignment with commercial firms. There may be tensions between developers considering being part of the free software community and developers considering being part of the open source community.
- ***Personal hostilities:*** Personal hostilities and clashes based on different characters is an important reason for forks. Core developers sometimes simply do not get along with each other.
- ***Commercial:*** There may be commercial interests to fork a project to appropriate profits. By doing so a new license can be chosen that is more suited for commercial purposes. A project under a BSD license can for example be relicensed under a proprietary license. Forking a project also may be done for gaining control over project development. Thereby the software can for example be made compatible with the own products and incompatible with competing products. While forks of copylefted software are possible the commercial incentives to do so are considerably reduced (Wheeler 2005). Source code changes cannot be kept private if the software is distributed. This allows competitors to incorporate these changes into the own project at no charge. Therefore, it makes no sense to fork a project under a GNU GPL license in order to being technologically ahead of competitors.

2.7.3 Forking and the FOSS phenomenon

Interestingly, in FOSS history there were many BSD forks producing the FreeBSD, NetBSD, OpenBSD, and DragonFly BSD projects but few or no relevant forks of the Linux kernel.[122] Young and Rohm (2000: 247-243) argue that the license terms of the GNU GPL are responsible for this observation. Because derivative work cannot be appropriated under a GNU GPL license there is less commercial motivation to fork than under a BSD license. Because everybody can draw from the source code the GNU GPL prevents forking (Torvalds & Diamond 2001: 169-170). There is no reason for commercial firms to bear the costs of forking without being able to appropriate the advancements.[123]

There is a desire to not split a project into competing forks (Feller & Fitzgerald 2002: 96). Development capacity is dispersed into different projects which slows FOSS advancement in its entirety down. Behlendorf (1999: 161-162) concludes:

> *In fact, the amount of talented programmer effort available for a given set of tasks is usually limited. Thus, it is usually to everyone's interests if parallel development efforts are not undertaken simply because of semantic disputes between developers.*

Wheeler argues that the large number of Unix forks is one of the main reasons that Unix systems lost considerable market share to Windows. That forking is relatively uncommon (Raymond 2003, fork) in FOSS development is perhaps the result of the Unix forks. These forks were such a bad example that the community refrains from forking.

For Weber (2004: 180) the right to fork source code transfers a very important resource of power from leaders to followers (see the Mambo-Joomla! example below). Wheeler argues that a fork is for the same reason

[122] An example from the Linux kernel is the fork of the glibc library by the Linux libc library.
[123] However, at least some of the mentioned BSD forks had clearly no commercial reason.

important as a motion for confidence in parliaments. The ability to create forks forces project leaders "to pay attention to their constituencies". The threat to fork is sometimes enough to cause project leaders to pay attention to an issue ignored so far.

2.7.4 The Mambo-Joomla! example and the power of forking

An impressive example of the power of developers is the Mambo-Joomla! fork. Mambo is or probably was a known and very successful content management system winning many awards such as "Best Open Source Solution" and alike (Linux News Desk 2005). Owner of the trademark and some of the copyrights was Miro International, a commercial firm selling products and services related to the Mambo software. In August 2005 Miro International started to reorganize the project. However, concerns about the decision making structure alarmed the regular project members (Vaughan-Nichols 2006):

> *We, the development team, have serious concerns about the Mambo Foundation and its relationship to the community. We believe the future of Mambo should be controlled by the demands of its users and the abilities of its developers.*
>
> *The Mambo Foundation is designed to grant that control to Miro, a design that makes cooperation between the Foundation and the community impossible.*
>
> *The Mambo Foundation was formed without regard to the concerns of the core development teams. We, the community, have no voice in its government or the future direction of Mambo.*
>
> *We don't see this issue as black and white Mambo v. Miro, {but} as we've said on the site, we don't believe the establishment of the Foundation was in the spirit of open-source software and its development.*

The statements above make clear that the new decision making structure did not correspond with the core values of the developers. Because Miro International did not respond adequately to the concerns the entire team of core developers left the project.[124] Based on the source code of the Mambo project they started a fork called Joomla!.

2.8 Development phases in FOSS projects

The traditional model for software creation is the waterfall development model. In this model development moves through different phases of activities such as analysis, coding, and testing (Messerschmitt & Szyperski 2003: 70-73; Royce 1987: 329):

[124] As it seems the users followed the developers. The official Joomla! Forum alone had in January 2007 47'070 new posts (Joomla! 2007) while the official Mambo forum had in the same period 287 new posts in total (Mambo Foundation 2007).

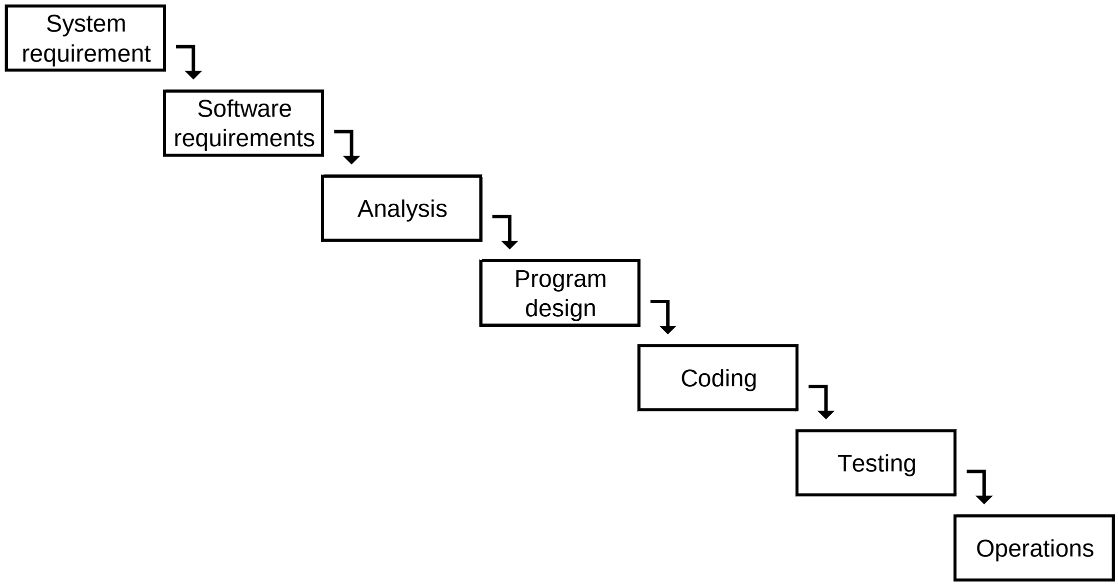


Figure 32: The waterfall model: A „grandiose approach"[125]

Funnily enough Royce (1987: 329) who is thought to be the father of the waterfall model describes it as risky and an invitation for failures. He used this model for software development only as a *bad reference* for suggesting a more iterative development process. Royce (1987: 325) pleads for example for an extensive customer involvement because "to give the contractor free rein between requirement definition and operation is inviting trouble".

Generally, FOSS development does not follow the process steps defined in the waterfall model. Rather, it follows an iterative development process (Jorgensen 2001: 321) as for example specified in the spiral model (Boehm 1988; Messerschmitt & Szyperski 2003: 74-75).[126] Software is then developed incrementally (FTC 2003: 44, chapter 45; see also Merton 1993) and each iteration involves modifications of existing functionalities while new ones are added. Similarly to Agile methods of software development such as for example Extreme Programming feedback is central for the iteration process in FOSS development (Jeffries 2004: xxiv).[127] While this iterative mode of development is on the one hand very adaptive to changes of user requirements it produces on the other hand some problems for documentation (Berglund & Priestly 2001: 132).

In FOSS development there is usually no scheduled project end. Projects are developed further as long as there are enough programmers willing to contribute. In consequence, FOSS development phases cannot be categorized along activities but along development phases or project maturity. Most hosting platforms such as SourceForge, Freshmeat, berliOS, or GForge use a categorization similar to the following one: Planning, pre-alpha, alpha, beta, production/stable, mature, and inactive. As can be see in Figure 33 the projects at the hosting platforms SourceForge, Freshmeat, and berliOS are more or less evenly distributed over the phases planning, pre-alpha, alpha, beta, and production/stable. There are only few projects in the category mature.

---

[125] Royce (1987: 329).

[126] For an interesting overview over lifecycle models see McConnell (1996: 156-157).

[127] A major difference between FOSS development and Agile methodologies is that in the later case face-to-face feedback is very important (see also the Agile Manifesto of Beck et al. 2001; Goldman & Gabriel 2005: 41).

Project development phases							
	Planning	Pre Alpha	Alpha	Beta	Production / Stable	Mature	N
SourceForge[a]	21.57%	16.85%	17.71%	22.99%	19.23%	2.31%	89'736
Freshmeat[b]	0.29%	3.51%	14.19%	32.35%	43.70%	5.96%	28'396
berliOS[c]	20.40%	18.88%	19.17%	23.05%	16.96%	1.55%	3'167

[a] 15.06.2006
[b] 11.07.2006
[c] 22.09.2006

Figure 33: Development phases of FOSS projects[128]

Schweik and Semenov (2003) identify three phases in the lifecycle of FOSS projects. There is i) project initiation, ii) going open, and iii) project growth, stability or decline. In the project initiation phase decisions related to design and modularity have to be made. In the going open phase a license has to be chosen. Additionally, project credibility, adequate communication systems, suitable version control systems, effective recruitment strategies, and appropriate project governance structures and institutional designs (e.g. peer review system or a mechanism for conflict resolution) have to been built up. The last phase is dominated by recruiting developers and users for testing and bug reporting. Interestingly, Schweik and Semenov (2003) argue that "some of the motivations for joining an OS project … are the same as for initiating one, but there are some new motivations as well."

2.9 Economic characteristics of FOSS applications

Discussing the economic characteristics of FOSS applications is of special interest because by defining FOSS as a public good a link to a basic economical problem (see chapter 5) is established.

2.9.1 FOSS applications are information goods

To define information is not an easy task.[129] However, a very simple but also very sound definition is that information is anything that *can* be but not necessarily is digitized (Shapiro & Varian 1999: 3). Based on this definition software and therefore FOSS is an information good (Messerschmitt & Szyperski 2003: 268).

An important difference between information and material goods is that in the former case the benefits cannot be totally transferred (Fulk et al. 2004: 570). The reason why information cannot be transferred completely is that the individual will still have access to it and continue to benefit from it. Information goods are reproduced if they are transferred. Ownership of material resources or money can be transferred from an actor to another actor in such a way that the former owner has no access to the good anymore.

Information is to some extent non-rival because its consumption does hardly diminish its availability for use by another individual (Benkler 2002: 369; Schmidt 2006: 125). Thomas Jefferson (1813, image 1062) wrote:

[128] BerliOS (2006), Freshmeat (2006), and SourceForge (2006b).

[129] For a definition of information see Messerschmitt (2003: 14-20). See also Johannessen et al. (1997: 153) who define information as "the difference which makes the difference".

If nature has made any one thing less susceptible than all others of exclusive property, it is the action of the thinking power called an idea, which an individual may exclusively possess as long as he keeps it to himself; but the moment it is divulged, it forces itself into the possession of every one, and the receiver cannot dispossess himself of it. Its peculiar character, too, is that no one possesses the less, because every other possesses the whole of it. He who receives an idea from me, receives instruction himself without lessening mine; as he who lights his taper at mine, receives light without darkening me.

Information is an experience good because the quality and utility is best judged by consuming it (Messerschmitt & Szyperski 2003: 20; Shapiro & Varian 1999: 5-6). Therefore, if information is not contributed to a common it is relatively invisible. It is hard to figure out which information an employee does not share with his coworkers if they have not consumed it yet. For information commons free riding is therefore difficult to detect. Contributions to common resources in form of money are more easily observable. In consequence, a mandatory system for contribution to information commons may be troublesome to install (Fulk et al. 2004: 571). The absence of a reliable mechanism to monitor and control information quality may lead to a system with low quality (Kalman et al. 2002).[130] A related characteristic of information is that not only a Leviathan enforcing cooperation may have difficulty to judge the contribution of an individual to the information commons but also the individual himself. The quality of contributed information is difficult to judge for the contributor because information goods are instable in their value across different settings and time (Fulk et al. 2004: 571).The marginal cost of the reproduction of information are, in contrast to the production costs, very low (Fulk et al. 2004: 571; Messerschmitt & Szyperski 2003: 2; Välimäki 2005: 51). Communication and the exchange of information across time and space is cheap (Benkler 2002: 404). This is even truer for FOSS because it is exchanged mostly over the Internet. For commercial software manuals have to be printed and there are costs for selling the software. There are therefore huge economies of scale for information, for software (Messerschmitt & Szyperski 2003: 323), and especially for FOSS. Behlendorf (1999: 151) argues similarly that software exhibits dramatic economy of scale and that there is almost no “direct correlation between the amount of effort it takes to produce a software product and the number of people who can thus purchase and use it.” At the same time there are probably little diseconomies of scale (see also Williamson 1975: 126-129; Zenger 1994) for software in general and for FOSS specially. Other characteristics of information goods are that i) they decay during time and must be regularly updated and that they are ii) valued differently by different individuals.

Benkler (2006: 3-4) discusses a shift from the industrial information economy to the networked information economy. In this economy decentralized individual action dominates and requires new modes of cooperative and coordinate actions. Benkler (2006: 5) argues that the emergence of FOSS and other goods based on peer production such as Wikipedia is the result of this shift. A major driving force behind this shift is the declining price for communication, computation, and storage. In the networked information economy the physical constraints are partly removed and with it some of the financing problems. Building premium software without big financial resources is possible as long as bright programmers contribute source code in their spare

[130] The following real example came to the author’s knowledge: A big group of call agents was expected to assist customers by phone. The call agents had i) to assist customers respectively solve their problems and ii) to enter the solution into a database which was thought to help other call agents. The time a call agent was online helping customers was easily measured by seconds. However, the quality of his entry to the database was very difficult to control. Because call agents were heavily urged to be as much on the phone with customers as possible the entries into the database were made as fast as possible. Within very short time the quality of the database degenerated. The database got useless and apart from entering data call agents stopped very fast using it.

time. The same mode of production is for example hardly possible for building a communication satellite. Somebody has to pay for the rockets that boost the satellite to a geosynchronous orbit.

2.9.2 FOSS applications are public goods

The characteristics of a public good are that nobody is excludable and that there is no rivalry in consumption (see 5.3 for definitions). These two characteristics will be discussed in turn in relation to FOSS.

Non-excludability

There are different argumentations why FOSS has to some extent an excludable and to some extent a non-excludable character. On the one hand one may argue that FOSS licenses grant (see chapter 2.5.1.2) the freedom to redistribute and modify the source code to anyone adhering to its terms. Basic to the FOSS philosophy is that there should be no discrimination against anybody. The open source definition states that open source licenses must comply inter alia with the following criteria: No discrimination against persons, groups, or fields of endeavor (OSI 2006b). It can therefore be argued that FOSS is a non-excludable good.

On the other hand subjects who do not comply with the terms of a FOSS licenses are excluded from using the software. Section 4 of the GNU GPL states (FSF 2006c):

> *You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License.*

That individuals who do not comply with the license terms are excluded is the very important difference between software under a FOSS license and software in the public domain. One may therefore also argue that FOSS is partly an excludable good. Similarly, von Hippel and von Krogh (2003) suggest that FOSS is partly an excludable good that should be explained by a private - collective model. They argue that some benefits associated with contributions to FOSS can only be enjoyed by project members. Significant private elements such as fun and learning are not available to free riders (von Hippel & von Krogh 2003: 216). In fact, contributors get some sort of selective incentives. Von Hippel and von Krogh argue further that free revealing of contributions to the FOSS community does not necessarily represent a loss of private profit for the contributor. Free revealing can increase the diffusion of the FOSS project and thereby increase related profits. A developer investing in his reputation benefits if more individuals “free ride” and use the software even if they do not contribute anything. The same argumentation may hold true for developers selling consulting for software.

It can be summarized that FOSS is to some extent a non-excludable good. However, some selective incentives are only accessible to contributors.

Non-rivalry in consumption

Digital information is non-rival in use (see also chapter 2.9.1). An indefinite number of users can be given replicas of information without interferences (Messerschmitt & Szyperski 2003: 20). The use of software by an individual does almost ever not diminish its value to another individual. On the contrary, in many cases the value of software increases with its use.

There are three main relationships of an individual to a FOSS project. *Firstly*, an individual does neither use the software nor contribute to its development. This is the least beneficial for a FOSS project because he contributes nothing to the improving of the source code nor does he contribute to a network effect (see chapter 2.9.3). *Secondly*, an individual uses the software but does not contribute. He therefore takes a free ride. In contrast to a regular common good free riders do not undermine production (Benkler 2002: 438). In fact, for some types of FOSS developers the amount of free riders not contributing to the public good FOSS respectively mere users increases the payoff associated with contribution. For example developers interested in reputation may be attracted by a project because it has many free riders respectively users. Every additional user contributes to the value of a FOSS project by fostering the network effect. Therefore, it is better if somebody is a free rider than he uses the software not at all. *Thirdly*, an individual uses the software and contributes to it as well. Certainly, this is the most beneficial relationship between an individual and a FOSS project. However, the vast number of users has not the skills needed to do so. It can be concluded that free-riding respectively only using software is ceteris paribus not negative but positive for the success of FOSS projects.

FOSS applications are public goods

As argued above FOSS is to some extent a non-excludable good with non-rivalry in consumption. It can be concluded that FOSS is not a pure public good. However, as discussed in chapter 5.3 the conditions qualifying a good as a public are seldom completely satisfied. For many goods non-rivalry in consumption and non-excludability are only satisfied to some degree. Based on the argumentation above it seems to be appropriate to consider FOSS applications as public goods.

2.9.3 Network effects, switching costs, and lock-in effects

North (1997: vii) notes that “history matters” and “today’s and tomorrow’s choices are shaped by the past”.[131] Institutions and standard constraints of economic theory determine the opportunities for organizations that in turn are created to take advantage of these opportunities (North 1997: 7). Thereby these organizations alter institutions. The resultant path of institutional change is then shaped by a lock-in that comes from the symbiotic relationship between institutions and organizations that have evolved as a consequence of the incentive structure provided by those institutions.

In the following, different theoretical models about the above introduced phenomenon are shortly discussed. These models are afterwards partly integrated into a larger framework. Finally, the described phenomenon is connected to software in general and to FOSS specifically.

[131] An interesting example how the past influences the future is the Matthew effect coined by Merton (1968a; 1973: 439-459). Distinguished scientists get often more credit than a comparatively unknown researcher does even if quality of their work is similar.

2.9.3.1 Lock-in effects and the like: Theoretical models

Increasing return technologies

According to Arthur (1989: 116) modern and complex technologies display often increasing returns to its adoption. The more people use a technology the more is learned.[132] The value of a technology increases in this case with its adoption. Arthur coined technologies with such characteristics as "increasing return technologies". If two or more technologies compete for a market of potential adopters the technology that gains an early lead in adoption (e.g. technology A in Figure 34) may lockout other technologies (e.g. technology B in Figure 34). Therefore, it can be concluded that history matters (Arthur 1989: 127-128).

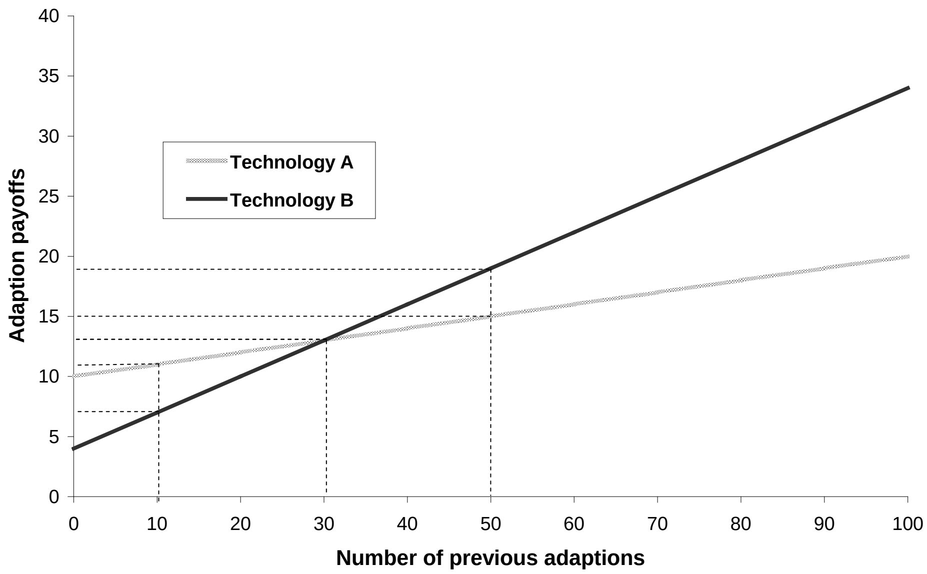


Figure 34: Example of a path dependency[133]

A real example of an increasing return technology may be nuclear reactor technology (Arthur 1989: 126). The reason why the U.S. nuclear industry is dominated by light water reactors (CEC 2006) can be partly attributed to the adoption of this technology by the Navy for the first nuclear submarine USS Nautilus. Related learning and construction experience respectively the huge sponsoring of the technology locked in the industry although other reactor types may also have been promising for regular energy production.

Path dependencies

There are different definitions of path dependencies (Liebowitz & Margolis 1995: 206-208). In a widely agreed definition initial actions put individuals respectively customers on a path that cannot be left without facing costs for doing so. [134] At the same time another path is more beneficial in the end. For example the ability of a

[132] The source of increasing returns is learning and not network effects.

[133] Arthur (1989: 119).

[134] However, we know from disruptive technology (Christensen 1997) respectively disruptive innovation (Christensen & Raynor 2003) that there are low-end and new market disruptions. In the former case customers are targeted who do not need full technological performance. In the later case a technology is inferior in most performance measures but

firm to adopt a new and promising technology is a function of its level of related experience with an old technology (however, see Cohen & Levinthal 1990: 135-138; Zhu 2001: 521). The more experience a firm has with a technology the more has been learned and hence the more costly it is to switch to another technology.

It may be that the payoffs of the different paths are not known when one has to decide. This is the case if technological advancement cannot be specified. However, it also may be that the payoffs are approximately known and that it is understood that the chosen path is inferior in the long rung. In such a case the inferior technology may be chosen nevertheless. For early adopters the payoff for using the inferior technology may be higher. In the example presented in Figure 34 the very first customer gets a payoff of 4 for choosing the superior technology B and a payoff of 10 for the inferior payoff A. Rational egoists clearly choose A although this lead to lower payoffs for others. Therefore, that the disadvantages and the possibility of a lock-in are known will not prevent that a clearly inferior technology prevails in the end.

Switching costs

Switching costs are costs that emerge if an actor migrates from one product to another (Välimäki 2005: 56-57). If such switching costs are high enough it affects the further purchases insofar as the product already in use is favored. In consequence, a lock-in situation emerges. Switching costs are based on minimum order-size commitments, loyalty programs, durable purchases [135], supplier specialization [136], brand-specific training respectively learning costs, information and database file formats, search costs, and social respectively psychological costs (Farrell & Klemperer 2006: 12; Klemperer 1995: 517; Shapiro & Varian 1999: 117).[137] In either case switching costs are based on decisions made in the past.

Network effects

Positive network effects arise when the payoff for individuals from an adoption of a product, technology, or service increases with the number of other individuals who adopt it as well (Katz & Shapiro 1985: 424).[138/139]

superior in an emerging market segment ignored by established firms. An example of a new market disruption is the floppy disk. While the 5.25-inch floppies were superior to the 3.5-inch floppies concerning storage capacity, the later were smaller and had a better protection against unintended physical contact. In consequence, a technological lock-in may be escaped if there are either low-end or new market disruptive technologies respectively disruptive innovations.

[135] Expensive products accompanied by complementary products and services may be very durable and sometimes not disposable. In this situation it is costly to switch the product (Shapiro & Varian 1999: 118-121).

[136] A supplier may gain some special knowledge about a certain customer that makes switching very costly (Shapiro & Varian 1999: 123-126). Customers with very costly and specialized demands such as the Pentagon or the NASA find that the losers in the bidding for a complex product cannot maintain the specialized expertise without ongoing business. In such a situation a customer may dual source. It this is not possible because of large fix costs organizations may find it worthwhile to fund its contractors to maintain their capabilities to bid on the next major contract.

[137] There are also switching costs for firms for serving new rather than old customers (Klemperer 1995: 518-519; Shapiro & Varian 1999: 130-131).

[138] There are also negative network externalities (Liebowitz & Margolis 1994: 134). However, it is thought that they are not relevant for the argumentation in this study and in consequence they are not discussed here.

[139] According to Metcalfe's Law the value of a network increases proportionally to the square of the number of users. This implies that the individual user's utility is a linear function of network size (Swann 2002: 418-419). While the fundamental insight that the value of network increases with the number of its users is correct the quantitative statement of Metcalfe's Law seems to be incorrect. It overstates the value of large networks (Briscoe, Odlyzko, & Tilly 2006; Rohlfs 2001: 29-30 and 195).

Widespread adoption of products, technologies, or services causes value. The value of having a telephone for example increases with the number of other individuals who have a telephone too.

Such network effects arise inter alia if users want compatibility with other users so that they can interact with them (Farrell & Klemperer 2006: 6). Network effects create network externalities for individuals. Such externalities respectively the consequences of an activity are not reflected in market prices. An externality is a situation where the action of an actor affects others without compensating payment (Messerschmitt & Szyperski 2003: 52). For purchasing a phone a consumer is not paid by other consumers although he makes their phones more useful.

According to Katz and Shapiro (1985: 424) there are several sources of network effects:

- ***Direct effects:*** There are direct effects of the number of purchasers on the quality of a product. An example is the already mentioned communication network for phones.[140/141] In Figure 35 a network with incompatible phone sets is presented while there is in Figure 36 a phone network with compatible phone sets. Although there is the same number of phones in both networks their usefulness is different. In the case of the network with incompatible phone sets the two customers have only one person they can contact while in the case of the compatible network each customer can call three other customers.
- ***Indirect effects:*** Indirect effects influence the value of a product. An example is computer hardware for which the amount and variety of software increases with the number of hardware units sold.
- ***Postpurchase services:*** The quality and availability of postpurchase services is a function of sold units. In the automobile market for example the service network depends on the number of sold cars.
- ***Information:*** Information about a good is more easily available for goods that are more popular. An example may be a heavily visited restaurant for which many quality reports are available.
- ***Signal of product quality:*** The market share may serve as a signal of product quality. It may be inferred that an always fully booked craftsmen performs high quality.
- ***Psychological effects:*** Due to purely psychological effects individual may prefer being customer of a highly adopted good.

[140] In fact, the usefulness of a communication network should be computed by taking into account the number of connections as well as the importance of each single connection. As we know from history the famous “red phone”, initially not a phone but a telegraph-teleprinter, was very important during the Cold War despite connecting only the White House respectively the Pentagon with the counterpart in the Soviet Union (BBC News 2003). It was established after the Cuban Missile Crisis in which the two countries came close to mutual annihilation. The two countries had big problems to communicate with each other. At the tensest moments of the crisis Soviet ambassador Anatoly Dobrynin had to rely on a bicycle courier to pick up his urgent messages to Moscow and bring them over to the local Western Union office (Kennedy 2007). The simplification to take only the number of connection into account for describing a communication network seems to be nevertheless justifiable because not all communication is that important as the one between the US and the SU during the Cold War.

[141] The red phone is also an informative example how communication fosters cooperation (see chapter 5.4.2.1.2). Not only the possibility to communicate itself was tremendously important but also the jointly process of negotiating a memorandum of understanding that installed the “red phone” (Jamgotch 1985: 9). The cooperation needed to install the “red phone” spilled over into other cooperative communication efforts.

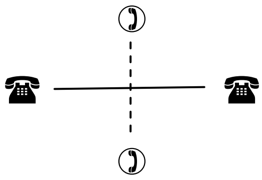


Figure 35: Network with incompatible phone sets

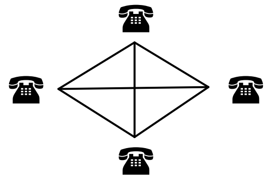


Figure 36: Network with compatible phone sets

Due to network effects there are incentives to herd with others (Farrell & Klemperer 2006: 10).[142] This leads to self-fulfilling expectations in which a network that looks like succeeding will do so as a result.[143] In such a case it is crucial for a technology, a project, or a product to reach fast a critical mass to get over the tipping (see chapter point 6.1). Very early adoption is then crucial for success (Arthur 1989).[144]

2.9.3.2 Bringing the different models together

Network effects lead to network externalities and decision made in the past respectively path dependencies to switching costs. It seems to be clear that network effects and path dependencies influence each other. Based on contractual conditions there may be high switching costs for users that in turn raise the number of users and may lead to network effects. A software firm may for example give early users a very high rebate if they agree being loyal. It also may be that an individual chooses a software solution because of the good opportunities to exchange files. In consequence, these users invest time to learn the use of the software that produces switching costs. Network externalities and switching costs together lead to a lock-in effect in which it is difficult for customer to change the product (see Shapiro & Varian 1999: 117).

Network effects	→	Network externalities		
⇅			→	Lock-in effect
Path dependency	→	Switching costs		

Figure 37: Model overview of lock-in effects and the like

[142] Liebowitz and Margolis (1994: 134) argue that the reason to live in an urban area is that some activities function best or only if a large number of participants are involved. Anonymity may be an example.

[143] Self-fulfilling prophecies have been described by Thomas (1928) and Merton (1948). An example of a self-fulfilling prophecy is the foresight given to Macbeth by three witches (Shakespeare 1992b: 12). Being told that the gets king Macbeth tries to fulfill the prophecy that thereby ends up coming true. Self-fulfilling prophecies have been prominently discussed for financial markets (Dellas & Stockman 1993; Obstfeld 1996; Weber 1998). The classical example for a self-fulfilling prophecy is Merton's (1968b: 476-477) sociological parable of the fictitious "The Last National Bank". This bank was a flourishing and solvent institution. Like any other bank most liquid assets were invested. However, at a Wednesday several customers, most probably laid of workers of a factory who needed cash, started to withdraw money. Because so many people withdrew their money a false rumor about insolvency started to flourish. Cautious customers started to withdrew their money as well. Thereby the rumor appeared to be true to the common customers and a rush to the bank started. Not prepared for such a concentrated withdrawal the bank got finally insolvent.

[144] Arthur (1989) stresses that insignificant and random events may give a certain technology an initial advantage.

The false QWERTY example

A classical example of switching costs and network externalities is the QWERTY typewriter keyboard. David (1985; see also Olson & Jasinski 1986) suggested that this type of keyboard remains dominant despite of its inferiority. For an individual switching to a superior keyboard design such as the Dvorak Simplified Keyboard is too costly. *Firstly*, this individual has to relearn how to type. *Secondly*, knowing how to type on a keyboard design nobody else use may lead to problems. On the hand it is difficult to buy an appropriate keyboard and on the other hand standard keyboards provided by firms cannot be used. [145] It is certainly beneficial for everybody if keyboard design is the same regardless of the keyboard producer.

Although the QWERTY example seems to be very plausible Liebowitz and Margolis (1990) showed clearly that the example is wrong. The data on which the claims of the superiority of the Dvorak Simplified Keyboard are based on is at best suspect. The most dramatic claims for the superiority are traceable back to Dvorak himself. Independent ergonomic studies on keyboards suggest that the QWERTY is not inferior. Liebowitz and Margolis (1994: 149) state that the QWERTY example is wrong. They further argue that some phenomena that look like network effects are simply manifestations of technological progress. Although the QWERTY example seems to be wrong it remains instructional.

2.9.3.3 Switching costs and network externalities for software and FOSS

Switching costs and network externalities for software in general

Switching costs and network externalities are central to the "new economy information technology industries" (Farrell & Klemperer 2006: 6). Software and specially FOSS are an "increasing return technology" as described by Arthur (1989: 116).[146] This may explain why in the software industry typically only few products dominate the market at a time. It is argued that for software the following network externalities are relevant:

- ***Direct effects:*** There are huge direct network externalities for software (Messerschmitt & Szyperski 2003: 52). For many types of software an exchange of files is crucial for its use. This is even truer in the Internet age in which the costs for file sharing got very low. This gives widely used software a heavy advantage over less popular solutions. Everybody converting word processing files from one software solution to the other knows that it is still not functioning properly. The problem of import and export of data does even exist between different products of the same firm. One important aspect for success of the open file format PDF is probably that it can be easily exchanged without major problems.

  Another direct network externality is that the more widely distributed a software is, the more opportunities for using related knowledge exists. If you have learned at your home computer how to use the operating system Windows you can use this skill at most other computer terminals. In contrast, if you use Linux there are fewer opportunities to use your corresponding knowledge.

[145] This argument only holds true for typewriters but not for computers. For computers it is hardly a problem to adjust the keyboard design to personal preferences.

[146] There are technologies that do not enjoy increasing returns with adoption (Arthur 1989: 117). An example is hydroelectric power that becomes more costly with adoption because potential dam sites become scarcer and less suitable.

Microsoft seems to have used the power of direct network effects to outcompete competitors. The U.S. Department of Justice alleged Microsoft to follow a “embrace, extend and extinguish" strategy (DOJ 1999: section V, 91.93). According to this incrimination Microsoft produces software substantially compatible with competing products or open standards (embrace), adds new proprietary features not supported by the competing product or the open standard which creates heavy interoperability problems (extend), and use the market power to marginalize competitors (extinguish) which cannot support the extensions (Economist 2000).

- ***Indirect effects:*** For software there are also positive indirect externalities (Messerschmitt & Szyperski 2003: 53). The development costs of software do hardly depend on the amount of users.[147] At the same time the software reproduction costs are very small. Because fixed costs are much higher than variable costs there is an incentive to develop software only for the largest operating system.[148] Microsoft’s operating system for example attracts a lot application programming because it is widely used which in turn is a self-enforcing process.
- ***Quality increases:*** Much prominent software is tested, intentionally or unintentionally, by its users. Microsoft for example offered developers to test Windows Vista. Afterwards, there was also a free download for all users to test the operating system. In FOSS the incorporation of users does even go further and is a prominent part of the production model. For a FOSS user it is not only possible to report a bug but also to fix it.

Next to the network externalities discussed above the following switching costs seem to be relevant:

- ***Loyalty programs:*** Previous software customers get very often special discounts for upgrades. This can be interpreted as a loyalty program. A customer changing the software solution has to pay the full price of new proprietary software.
- ***Durable purchases:*** While *installed* software or the data storage device itself (see for example Slattery et al. 2004) may degenerate over time. However, the *binary code* respectively the *source code* is infinitely durable. If installed software is degenerated it can be reinstalled. While a bouquet of flowers outlive a few days and a car a few years the absolute performance of software remains constant.
- ***Brand-specific training / learning costs:***[149] People have to learn to handle interfaces, keyboard shortcuts, system crashes, and the like. Gaining some familiarity with new operating systems needs some time even it is not more complicated than the old one. That is probably why software companies often only charge symbolic rates for schools and universities.[150] Once students are accustomed to a certain solution they will not easily change to another solution if they leave school. If employers use this software they do not have to familiarize the employees with the software solution.

---

[147] This is obviously not completely correct. If there is only one user with known hardware, operating system, consumer need, and the like development is less complex and therefore cheaper.

[148] However, the interoperability problem of applications between different operating systems is lowered by the rise of the Java Virtual Machine that allows writing an application once and running it on different operating systems.

[149] In the section about network externalities the use of software related knowledge was discussed as well. However, there the discussion was about the benefits of using this knowledge. Here the discussion is about the costs for acquiring the knowledge independent how it can be used.

[150] This is probably also why software products look similar to products of the market leader. Microsoft Excel looks very similar to VisiCalc; the first spreadsheet product and application market leader in the very late seventies and in the very early eighties. In both applications the row columns are labeled A, B, C across the top and 1, 2, 3 down the side.

- ***Information and database file format:***[151] Information and databases are stored in a certain file format and there may be difficulties porting them to other software (Shapiro & Varian 1999: 122-123, see also direct network externalities above). Some of the stored information may be even lost if the software solution is switched. Such problems are diminished if there are standard file formats and standard interfaces.
- ***Search costs:*** Before one switches to another product one must compare the different alternatives. This is a costly endeavor.

It can be concluded that there are clearly important network externalities and switching costs respectively network effects and path dependencies for software in general (Messerschmitt & Szyperski 2003: 310-320; OECD 2004: 25).[152] The larger the user base of a software is, the higher is the value of the product (Välimäki 2005: 57). Based on documents of the antitrust case U.S. vs. Microsoft (Bresnahan 2001) it seems to be clear that software firms or at least Microsoft are fully aware of the power of lock-in economics and network theory.

Switching costs and network externalities for FOSS software

The network externalities discussed for software mostly apply also to FOSS (Välimäki 2005: 56).

The value of a certain software solution increases with the possibility to use the knowledge about the application. It is also clear that more applications will be developed for an operating system with many users than for an operating system with few users. A FOSS project makes his application more likely suitable for Linux than for the lesser known operating system Plan 9. That quality increases with the number of users is even truer for FOSS because user involvement is an integral part of the production model. For FOSS there are ceteris paribus also the same search and learning costs as for proprietary software.

What applies clearly less to FOSS than to proprietary software are network externalities based on the exchange of files. FOSS projects have no incentive using special and strange file formats. The source code can be accessed anyway. FOSS projects therefore often use open standards.[153] In fact, open standards may lessen the importance of network effects considerably (see also Jakobs 2006). An example is the Internet protocol suite (e.g. TCP/IP or HTTP). Because it is an open standard users can run applications from different producers to share information. Even if a developer had built his own web browser from scratch he could browse the World Wide Web as long as he implemented the corresponding standards.[154]

There are no costs based on durable purchases and loyalty programs with FOSS because the software itself can be obtained for free.[155] It can therefore be concluded that the costs inherently to switching to another software

---

[151] In the section about network externalities the problem of incompatible data files was discussed as well. That discussion referred to the exchange of files *across* different users. Here the migration of data from one to another software solution of *one certain user* is discussed.

[152] There is certainly software that does not share or not fully share the discussed characteristics. Nevertheless, it may be argued that such software is overall of minor importance.

[153] For a definition of an open standard see Zhu et al. (2006: 517) or Valoris (2003: 20).

[154] If the communication protocol were proprietary only browsers paying for the standard would be able to browse the network. Because users like to communicate with as many other individuals as possible there is an enormous advantage to use the standard everybody else uses. One can assume that only one single standard would prevail. The producer of this standard would heavily benefit from the monopoly and users would be locked in.

[155] This statement is legally not correct. One may charge as much as one want for GNU GPL'ed software (FSF 2006d). However, because the terms of the GNU GPL require that everybody may distribute the software as well (as long as he

solution also exist for FOSS. However, switching costs based on file formats, durable purchases, and loyalty programs do not apply to FOSS.

To sum up it may be concluded that there are also switching costs and network externalities for FOSS software but that they are probably lower than for proprietary software.

2.10 Market share and importance of FOSS

Market share is for software very important because there are strong network effects (Messerschmitt & Szyperski 2003: 310) and big switching costs. In consequence, the market share of FOSS applications has to be discussed. Furthermore, doing so shows that the discussion about the FOSS phenomenon is important.

A study commissioned by the European Commission concludes that FOSS applications are first, second, or third-rung products in terms of market share in several markets such as web servers[156], server operating systems[157], desktop operating systems[158], web browsers[159], email[160], and other ICT infrastructure (UNU-MERIT et al. 2006: 18).

An impressive example of an area in which FOSS dominates are operating systems for supercomputing. More than three-quarters of the 500 fastest supercomputers run Linux as an operating system (University of Mannheim, University of Tennessee, & NERSC/LBNL 2006a). Of the fastest twenty supercomputers only two do not run Linux.[161] According to a representative of the Lawrence Livermore National Laboratory that runs the fastest and fourth fastest supercomputer in the world Linux dominates the marketplace for high-performance computing (Lyons 2005).

Market shares of FOSS applications have seen a considerable growth. However, it must be noted that in some areas the results of different studies vary considerably depending on methodology and prominently on the sponsor of the study. In other areas such as for supercomputing the market share data seems to be very reliable.

2.11 Summing up

In the following important *and* for the further argumentation integral parts of the chapter are briefly summed up. The basic insights of this chapter are:

---

accepts the terms of the license) it is likely that somebody distributes the software for free. Therefore, the statement is practically sound.

[156] As already discussed in chapter 2.2 the Apache HTTP server has a market share of approximately two-thirds (Netcraft 2007, in January 2007).

[157] In 2004 the share of total server shipments and redeployments of Linux was 20% (UNU-MERIT et al. 2006: 32)

[158] The market share of Linux on the desktop is, depending on used measurement, between 2% and 5% (UNU-MERIT et al. 2006: 32).

[159] In January 2007 the global market share of the Firefox web browser was approximately between 11% and 13% (Wikipedia 2007c).

[160] For email servers as the backbone of email services the total market share of FOSS is 48%. FOSS products occupy the 2nd (Sendmail), 4th (Exim), and 5th (Postfix) rank in market share (UNU-MERIT et al. 2006: 33).

[161] The fourth fastest supercomputer runs AIX and the 14th fastest runs Super-UX (University of Mannheim, University of Tennessee, & NERSC/LBNL 2006b).

- In FOSS development there are two different communities: The free software and the open source community. The former is foremost concerned about the freedom of software while the later is primarily interested in the development model. > See chapter 2.3.2
- Approximately two-thirds of FOSS projects are licensed under the GNU GPL. > See chapter 2.5.2
- Commercial firms and developers who do not adhere to community norms and legal requirements are a threat to FOSS development. > See chapter 2.6.1
- The GNU GPL forces in practice developers and commercial firms not only to take but also to give back. This makes it necessary that individuals engage in rule enforcement. However, the importance of the GNU GPL is not restricted to its legal function. It serves also as a constitution. It is the core statement of the social structure of FOSS projects. >See chapter 2.6.2
- Software is developed incrementally. FOSS projects can be categorized according to the development phase they are in. Some authors argue that the motivation to initiate a FOSS project is not similar to the motivation to join a FOSS project. > See chapter 2.8
- Information goods are relatively invisible that makes it hard for a Leviathan or an individual to evaluate the level of cooperation. > See chapter 2.9.1
- FOSS can be considered as a public good. > See chapter 2.9.2
- For software there are very high switching costs and network externalities. Users and developers tend to be locked-in. > See chapter 2.9.3

This chapter introduces the object of the study, namely the FOSS phenomenon. It is thought to be important knowing the characteristics of this object in order to develop hypotheses. Two of the characteristics are especially important. *Firstly*, FOSS licenses are crucial. It is argued in chapter 8.2.2 (hypothesis 12) that the GNU GPL as the most frequent FOSS license respectively copyleft licenses play a major role for pro-socially motivated developers. *Secondly*, it is very basic for the argumentation in this dissertation that FOSS applications are public goods. Thereby a connection to the enormous knowledge concerning this important problem can be established.

3 Motivation: The initiation force for individual behavior

To be motivated means to be moved doing something (Ryan & Deci 2000a: 54). An individual who feels no impetus or inspiration to act is therefore unmotivated. Somebody who is energized or activated toward an end or activity is motivated. Motivation is the psychological force within an individual that determines the direction of the individual's behavior. While motivation is a psychological state the result of it is behavior. Motivation and behavior is then something distinct (Mitchell & Daniels 2003: 227).

As described in chapter 2.4 the majority of software developers participate in FOSS projects without being paid. Benkler (2002: 373) observes that developers participate in FOSS projects without following the normal signals generated by market based, firm based, or hybrid based models. There seems therefore to be a puzzle why developers contribute to FOSS projects. However, knowing the motivation of FOSS developers is crucial. Founding member of the Apache Group and former primary developer of the Apache HTTP server Brian Behlendorf (1999: 161) notes:

> *Satisfying developers is probably the biggest challenge to the open-source development model, one which no amount of technology or even money can really address. Each developer has to feel like they are making a positive contribution to the project, that their concerns are being addressed, their comments on architecture and design questions acknowledged and respected, and their code efforts rewarded with integration into the distribution or a really good reason why not.*

Clary and Snyder (1999: 156) ask "why do significant numbers of people engage in the unpaid helping activities known as volunteerism?" This universal question of Clary and Snyder can be adapted to FOSS development as well: Why do software developers participate in FOSS projects?[162] In this chapter it is attempted to answer this question.

Managers of software projects should not assume that the developers are similarly motivated as they are. McConnell (1996: 253) writes: "If you're a manager and try to motivate your developers the same way you would like to be motivated, you're likely to fail". He further argues that "to motivate developers, emphasis technical challenges, autonomy, the chance to learn and use new skills, and career planning – and respect their personal lives".

Motivation is multifaceted

If human beings were understood in fashion of a homo oeconomicus[163] respectively as a wealth maximizing and self-interested actors one would first look how an individual benefits materially from contributing to FOSS

[162] The importance of non-salary incentives is not restricted to FOSS development. A study conducted by Green (1989: 130) found that non-salary incentives are important in general for employees in the IT sector.

[163] The term homo oeconomicus emerged in response to john Stuart Mill's work (Persky 1995: 222). According to Persky (1995: 222) the use of the Latin term homo oeconomicus instead of economic man can be at least traced back to Pareto (1909: 17, the use of the Latin term in the original edition could not be verified).

projects.[164] However, according to the economist Alfred Marshall (1910: 22-24) man do not only strive for boosting earned money.[165] There is also the desire to earn approval and to avoid peer contempt that motivates human beings. A professional man or an artisan will be sensitive to the approval of those in the same occupation and care little of other people. Similarly, winning a race or to get powerful may be a strong motivation for working hard irrelevant of the related financial benefits (Marshall 1910: 23):

> *For just as a racehorse or an athlete strains every nerve to get in advance of his competitors, and delights in the strain; so a manufacturer or a trader is often stimulated much more by the hope of victory over his rivals than by the desire to add something to his fortune.*

Different individuals have different types of main motivation (Ryan & Deci 2000a: 54). A student may be highly motivated to do homework out of curiosity or interest. A different student may be motivated by the approval of the teacher, another student because he wants to learn a new set of skills to get a better job and yet another student learns because of the rewards he or she gets if a good grade is presented to the parents. In this example the amount of motivation does not necessarily vary, but the nature and focus of the motivation surely does. Benkler (2002: 426-427) assumes that there are three types of motivations:

i. Monetary rewards

ii. Intrinsic hedonic rewards experienced form taking an action

iii. Social-psychological rewards such as peer recognition

Defending an accused in court may therefore be done for the fee, for the pleasure of creating an argument, or for the respect of the legal community. For Buchanan (1978: 366) there are two dimensions of human behavior. Human behavior is a mix between the motivational force of economic self-interest and what he calls community.

Feller and Fitzgerald (2002: 137-154) proposed an interesting framework to research the motivations of FOSS developers. They consider three broad motivational areas and two different levels of analysis important. The motivational areas are economics, socio-politics, and technology. These motivations are analyzed at the micro-level (individual developer) and at the macro-level (organization/community).

---

[164] Economists in contrast to other social scientists seem to be more interested in opportunities than preferences (Elster 1989b: 14-15). In some cases, this seems to be a sound approach. The fact that people in the aggregate buy less of a good when its price goes up can be explained independently of what motivates individuals to buy or not to buy. They buy simply less because they can afford less if their income remains stable (Elster 1989b: 15). Furthermore, opportunities are mostly easier to observe and change than preferences. Additionally, economists tend to focus on monetary rewards because they assume that i) individuals are willing to substitute non-monetary for monetary rewards and that ii) because money represents a generalized claim on resources that is preferred over an equal dollar-value payment in kind (Baker, Jensen, & Murphy 1988: 594-595).

[165] Interestingly, even the American Economic Association seems not to belief that individuals strive for maximizing money. To apply for membership one has to declare from out of three categories the own annual income (AEA 2006). Depending on annual income the member fee differs. It can be assumed that this self-declaration can neither be controlled because tax documents are not in all countries open to the public nor that ii) it will be ever controlled. It can also be assumed that the statement concerning income will not be made public. In this case it does not make sense to declare any income above the minimum as long as the maximization of money is the most important goal of an individual. Certainly, this argumentation does not hold up for well-known scholars because the might fear negative effects for their reputation if rumors about the misstatements spread by the administrative personal enter the scientific community. However, it can be assumed that most members of the AEA do not qualify for being the target of such rumors.

Motivational area	Level of analysis: *Micro level (individual)*	Level of analysis: *Macro level (organization/community)*
Technological motivations	To meet a personal technological need	To address the software crisis - particularly poor quality
	To exploit the efficiency of peer review, etc.	To share tedious developers tasks (testing, documentation) with users
	To work with "bleeding-edge" technology	To leverage the FOSS community for R&D
		To promote innovation
		To ensure transparency of the application
Economic motivations	To gain future career benefits	To exploit investor infatuation with FOSS
	To improve coding skills	To shift software from a commodity industry to a consumer-driven service model
	To "strike it rich" through stock options, etc.	To raise mind-share and strengthen brand
	Low opportunity cost - nothing to lose	To exploit indirect revenues - selling related products and services, accessorizing, etc.
		To make software affordable in developing countries
		To cut costs - cheaper platform than proprietary alternatives
Socio-political motivations	Ego gratification and signaling incentives	Social movement require an enemy - e.g. Microsoft
	Intrinsic motivation of coding	Overcomes "digital divide"
	Sense of belonging to a community	Ideology - software must be free
	Altruism	Model for wider domain - future model for work

Figure 38: Motivation for FOSS development[166]

Some motivations for FOSS development cited by (Feller & Fitzgerald 2002: 138-139) seem to be overestimated or overrated. What is called the software crisis (see for example Dijkstra 1972) will most probably not terminated by FOSS development.[167] Projects that adopt FOSS processes will still running over time and there will be low quality software that is hard to maintain. It is certainly not difficult to find a FOSS project that perfectly matches the mentioned defects. FOSS projects members such as Linus Torvalds who

[166] Feller and Fitzgerald (2002: 138-139).

[167] Dijkstra (1972: 861) argues that one reason for the software crisis is that programming capabilities do not cope with the huge increase in computer power and reliability. This does not seem to change fundamentally for FOSS development. Anyway, the discussion about a software crisis seems to be of minor value. A similar argument could be made for any other area of human life. If human capabilities to reach the wildest dreams, the biggest desires, or the most exceptionally visions are evaluated the result is always that they are insufficient.

strike it rich through stock options are probably that rare that such an argumentation can be considered practically irrelevant.

In the following subchapter only the motivation of the individual to contribute respectively the micro level is discussed in breadth. The macro-level and the motivation of organizations are only slightly touched.

Ideal types and mixes of motivational forces

It should be noted that the different motivations discussed in this chapter are ideal types. An ideal type as coined by Weber (1990: 10) is an abstraction that hardly exists in reality. An ideal type informs us how a theoretically constructed individual acts (Weber 1990: 4). Its use helps understanding a phenomenon and assists in advancing theoretical development. In reality, people are moved by an elaborate set of different motivations. Lakhani et al. (2005: 19) observe that in fact members of FOSS projects are motivated by a combination of intrinsic and extrinsic factors. Individuals join the FOSS community for a variety of reasons and no reason dominates the community (Lakhani & Wolf 2005: 16). Von Hippel and von Krogh (2003) suggest that FOSS development should in fact be explained by a private - collective model. They argue that contributions to FOSS projects include significant private elements such fun and learning that are not accessible by free riders. Therefore, the decision of an individual to contribute may be motivated by different motivational forces.

In an entertaining article Schelling (1980a: 98) describes the struggle of different inner forces as follows:[168]

> *… people can act as if they were not quite singular individuals with unique identities and values and tastes and memories and sensory perceptions that display smooth continuity over time. Maybe for some purposes each of us is like two or more different identities, not switching discontinuously as in amnesia or electric-shock therapy, but with enough oscillation to affect some … decisions …*

One of the many examples showing his point is Schelling's own attempt as a young boy to toughen him against cold temperature. He saw a movie about Admiral Byrd's Antarctic expedition and his habit to go outside shirtsleeves in order to prepare for the cold in the Antarctic. Schelling decided to do so as well and removed a blanket for sleeping. However, in the middle of the night he awoke cold and he retrieved the blanket. For the next days this game of going to bed with one blanket less and waking up in the morning with one more went on. The discoverer in Schelling and the inner force to sleep well fought over the question to sleep with or without the blanket.

Arrows (1950: 333) assumes that individuals may order all social states by whatever standards they consider relevant. A member of Veblen's leisure class might order the states exclusively on the criterion of his relative income standing in each while a believer in the equality of man might sort them in accordance with measures of income equality. While this assumption reduces complexity and therefore may be useful it is certainly not realistic as the example of Schelling shows. In reality, an individual may have struggles which course of action is appropriate. It is assumed that FOSS developers to some extent may order social state according to their preferences but certainly not perfectly.

---

[168] See Margolis (1984) for a very interesting approach that deals with two different competing types of utilities.

Consequences of behavior

Any interaction of individuals is associated with consequences that may be framed in costs and rewards (Thibaut & Kelley 1959: 12). Rewards may include for example the pleasure, satisfaction, or gratification a person enjoys. Costs may include physical or mental efforts such as embarrassment or anxiety (Thibaut & Kelley 1959: 13). The rewards and costs an individual receives for his decision do depend on:[169]

i. the motivational structure of an individual (this chapter)
ii. the circumstances and structure of the decision problem (chapter 4 and chapter 5)
iii. the decisions of other individuals (chapter 6)

The combination of the motivational structure of an individual, the circumstances and structure of the decision problem, and the decisions of other individuals may lead to special dynamics (chapter 7).

Although motivation, decision structure, and decisions of other individuals cannot totally be separated the corresponding discussion in this dissertation is organized in the following way: In this chapter motivation in general and specifically the motivation of FOSS developers are discussed. In chapter four the FOSS production model respectively the circumstances of the decision problem will be discussed. In chapter five social dilemmas respectively the structure of the decision problems will be addressed. Chapter six deals with the fact that decisions of other individuals may have a major influence on rewards and costs of individuals.

3.1 Self-determination theory

There are many theories related to motivation in the workplace and connected to organizational settings (for an overview see Donovan 2001). A very important concept is self-determination theory (Deci & Ryan 1985). It is on the one hand based on cognitive evaluation theory that specifies different human needs and on the other hand on organismic integration theory that specifies motivation in a continuum from intrinsic to extrinsic motivation (Deci & Ryan 1985; see also Ryan & Deci 2000b).

In the self-determination theory of Deci and Ryan (1985) there are two basic types of motivation: intrinsic and extrinsic motivation. The former refers to doing something inherently interesting and enjoyable while extrinsic motivation refers to something that is desirable in its outcome.[170/171] Intrinsic motivation is often clearly set

---

[169] Similarly, Milgrom and Roberts (1992: 25) distinguish between the coordination problem and the motivation problem.

[170] This definition of intrinsic motivation is different from the one sometimes used in economical writing. In such cases not only enjoyment based but also obligation based intrinsic motivation is part of the concept (see also footnote 171). For example Frey and Torgler (2007: 140) equate in their paper about tax morale and conditional cooperation intrinsic motivation with “the moral obligation to pay taxes or the belief that paying taxes contributes to society”.

[171] Lindenberg (2001: 331-336) separated intrinsic motivation into two different components: enjoyment based and obligation based intrinsic motivation. Enjoyment based intrinsic motivation is then built upon self-determined behavior without being forced by commands and without being paid (Lindenberg 2001: 319). Lindenberg (2001: 339) argues that obligation is *intuitively* the other form of intrinsically motivated behavior. The distinction of Lindenberg does not seem to be theoretically sound. Intrinsic motivation is based on the *activity* itself while extrinsic motivation is based on its *result*. One may feel obliged to follow an integrated norm although one does detest the activity itself. Environmentalists probably do not especially like the activity of collecting garbage in a wildlife sanctuary but do so because they think the result of doing so is valuable. The logic of self-determination theory would be very unnecessarily disrupted by the proposal of Lindenberg. In consequence, big theoretical as well as methodological problems would arise. The basic argumentation that intrinsic motivation as well as obligation-based motivation are important and therefore the later must be part of the former is clearly not a sufficient reason to abandon the inner logic of self-determination theory. Furthermore, after reading Lindenberg (2001: e.g. 335) one may think that self-

while extrinsic motivation is a broader concept and covers many different types of motivation (Ryan & Deci 2000a: 55). Extrinsic motivation can be either result of external pressure or an inner acceptance of the value of a task. An activity is extrinsically motivated if it aims at outcomes separable from the activity (Ryan & Deci 2000a: 60). Extrinsic motivation is therefore goal-instrumental (Weber 1990: 12) because action is solely a rational mean for success.

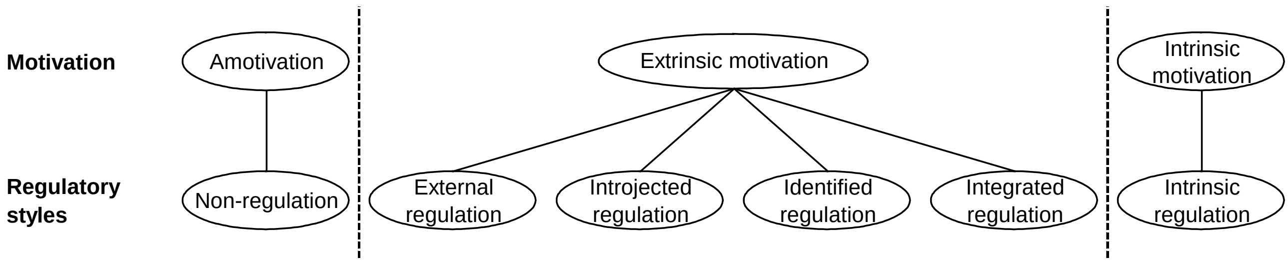
According to the self-determination theory extrinsic motivation can vary greatly in the degree to which it is autonomous (Ryan & Deci 2000a: 60). A student may do his homework because he fears parental sanctions. Alternatively, he may do so because he thinks that knowledge is a good thing for its own. In both cases the student intends to attain a certain outcome (no punishment respectively having knowledge) and action is in both cases instrumental. In both examples the student may consider learning (the activity) as painful. However, both types of extrinsic motivation vary in the relative autonomy. There is even extrinsic motivation without external pressure. In self-determination theory this is the case if values and regulations are internalized and integrated.

In self-determination theory there are the following regulatory styles for extrinsic motivation (see Figure 39):

- ***External regulation*** is the least autonomous form of extrinsic motivation. Action is performed to satisfy an external demand or to get an externally imposed reward (Ryan & Deci 2000a: 61).
- ***Introjected regulation*** describes a type of regulation that is based on guilt, anxiety avoidance, attaining of pride, or ego-enhancement (Ryan & Deci 2000a: 62). A person acts to enhance or maintain self-esteem and the feeling of worth.
- In ***identified regulation*** a person acknowledge the importance of a certain behavior. The regulation is then identified as his or her own (Ryan & Deci 2000a: 62).
- The most autonomous form of extrinsic motivation is ***integrated regulation***. It occurs if identified regulations are fully assimilated to the self (Ryan & Deci 2000a: 62). This includes that the new regulation is brought into congruence with the individuals' other values and needs. Integrated forms of motivations are very similar to intrinsic motivation. However, they are yet extrinsic because action is instrumental with respect to a certain outcome that is different from the action itself.

determination does not deal with obligation-based motivation that is clearly wrong. Integrated regulation is based on values and needs that individuals accept as their own (Ryan & Deci 2000a: 62). Such extrinsic motivation "share many qualities with intrinsic motivation, being both autonomous and unconflicted". Obligation based motivation is already incorporated in self-determination theory. There seems to be no sound reason changing the inner logic of self-determination theory.

Human motivation taxonomy by the Self-Determination Theory



Behavior	Nonself-determined	←			→	Self-determined
Perceived locus of causality	Impersonal	External	Somewhat External	Somewhat Internal	Internal	Internal
Associated regulatory processes	Non-intentional, nonvaluing, incompetence, lack of control	Compliance, external rewards and punishment	Ego-involvement, internal rewards and punish-ments, approval from self or others	Personal importance, conscious valuing, self-endorsement of goals	Congruence, awareness, synthesis of goals with self	Interest, enjoyment, inherent satisfaction
Motivation categories in FOSS projects		Salaried developers / Personal requirements / Reputation / Human capital	Peer recognition	Pro-social behavior		Fun seeker

Figure 39: Taxonomy of motivation by the self-determination theory[172]

3.2 Motivational forces in FOSS development

In the following different ideal types of motivation to contribute to FOSS development are presented. The discussion is organized based on self-determination theory. The different motivation types are aligned from intrinsic to extrinsic regulation.

3.2.1 Intrinsic motivation: Fun seeking

When Nobel Prize winner Albert Michelson was asked why he devoted so much time to measure the velocity of light he is said to have replied “it was so much fun” (Csikszentmihalyi 1991: 137).[173] Similarly, the Nobel Prize winner Arthur Schawlow said that the difference between a highly creative and a less creative scientist is that the former loves his labor, is driven by curiosity, and wants to know what the answer of a question is

[172] Ryan and Deci (2000a: 61; 2000b: 72).

[173] Albert Michelson (1887) is nowadays known for the famous Michelson-Morley experiment on the relation of luminiferous aether and the speed of light.

(reported in Amabile 1997: 39; Schawlow 1982). An important characteristic of Richard Feynman (see Feynman & Leighton 1997), yet another Nobel Prize winner[174], seems to have been his tremendous curiosity. He wanted to know how ants align, how safecracking works, how mind reading works, and how quantum electrodynamics works. Feynman (1982: 486) finished a keynote speech on simulating physics with computers that has been very influential for quantum computing with the words "by golly it's a wonderful problem, because it doesn't look so easy".[175] Similar stories could also be told about non-scientists such as the virtuoso cello player Pablo Casals or basketball player Michael Jordan (Amabile 1997: 39-40).[176]

What all the above-mentioned individuals have in common is that they are beside their success driven by intrinsic motivation. Intrinsic motivation can be defined as doing an activity because it is an inherent satisfaction rather than for some divisible consequence (Ryan & Deci 2000a: 56).[177] Individuals are sometimes motivated by having fun or being challenged. For such behavior there is no other reward than the behavior itself and there is no instrumental reason behind it (Ryan & Deci 2000a: 56).

The word intrinsic motivation seems to be first used by Woodworth (1918: 67). Intrinsic motivation is there related to absorption which means that the "attention is wholly directed upon the matter in hand" (Woodworth 1918: 69). He argues that where outside motives are necessary one cannot speak of absorption. One has to "get really into the subject, absorbed in it, finding it interesting and being carried along by the interest of it" (Woodworth 1918: 70). If extrinsic motivation from the outside interferes intrinsic motivation is distracted.

The idea behind the distinction of intrinsic and extrinsic motivation itself dates at least back to Aristotle (Schneider 1996: 124). Aristotle (1998: 257-260) writes that there is pleasure in an activity itself (intrinsic motivation) and pleasure coming from the outside of the activity (extrinsic motivation). He argues that having pleasure in the activity is beneficial for developing certain skills (Aristotle 1998: 258): "… those who enjoy geometrical thinking … grasp the various propositions better".

[174] Although it may be a coincidence, the found citations about intrinsic motivation of Nobel Prize winners refer all to physicists (Michelson, Einstein, Feynman, and Schawlow). However, an example of a mathematician intrinsically motivated is Grigori Perelman. For proving the Poincaré conjecture he was awarded the Field Medal that is thought to be the Nobel Prize for math. However, Perelman did not accept the price by the explanation that the Filed Medal is "completely irrelevant for me" and because "if the proof is correct then no other recognition is needed" (Nasar & Gruber 2006). He also announced to refuse a $1m prize offered by a private research institute (Randerson 2006). Additionally, Perelman was even not interested in publishing the proof in a peer reviewed journal.

[175] Actually, "Thank you" was the very last words of the speech (Feynman 1982: 486).

[176] Interestingly, Microsoft founders Bill Gates and Paul Allen searched at the age of 13 and 15 for bugs in DEC's computer program just for fun (Freiberger & Swaine 1984). For a long time they were not paid for their search. The Computer Center Corporation allowed Gates and Allen to use their computers during nights because as long as bugs were found the company did not have to pay the computers. Finally, DEC stopped this practice because they were scared that "these guys are going to find bugs forever".

[177] There is a wide variety of different definitions of intrinsic motivation. According to Heckhausen (1989: 455-460) there are six different conceptions of intrinsic motivation. Dyer and Parker (1975: 455) conclude that there is a lack of agreement concerning the definition of intrinsic motivation and found at least four different definitions. While Heckhausen (1989: 456) state that all definitions of intrinsic motivation have in common that behavior is not a mean for reaching a goal but rewarding itself, for Rheinberg (2006: 337) the different definitions do not have much in common. In a book on intrinsic and extrinsic motivation the editors state that the richness and diversity of the different contributions pose some problems and paradoxes (Sansone & Harackiewicz 2000: 8). Dyer and Parker (1975: 455) state also that this "plethora of definition" had led to inconsistencies. De Charms (1968: 328) also states that the terms intrinsic and extrinsic are poorly defined which may "cloud the central issue".

There is no activity that is per se intrinsically interesting or not (Ryan & Deci 2000a: 56). Some people are interested in some activities while others prefer different activities. It is therefore hard to tell whether an individual is intrinsically interested in a certain activity. Having said this it seems nevertheless obvious that there are activities appreciated by many individuals and activities appreciated by almost nobody. An example for the former case is perhaps playing a music instrument while an example for the later is probably cold calling (for a description of cold calling see Ronson 2006).

However, while in self-determination theory intrinsic motivation is an organismic propensity there are some catalyzers (Ryan & Deci 2000a: 58). There are conditions that do not cause but foster or hinder intrinsic motivation. If these conditions are favorable there is the possibility that intrinsic motivation flourishes. Nevertheless, intrinsic motivation does only occur for activities that hold intrinsic interest for an individual. According to cognitive evaluation theory there are three different basic human needs fostering intrinsic motivation (Deci & Ryan 2000: 228; Ryan & Deci 2000a: 58):

i. ***Autonomy*** is another basic human need (see also personal causation of De Charms 1968). Individuals must perceive an internal locus of causality respectively they must be self-determined.[178]

ii. The feeling of ***competence*** is related to having an optimal challenge, promoting feedback and freedom from demeaning evaluation (see also White 1959). Individuals must feel efficacious with respect to their activities.

iii. ***Relatedness*** refers to the desire to feel connected to others (see also Baumeister & Leary 1995). This includes to love and care and to be loved and cared for. Deci and Ryan (2000: 235) argue that there may be situations in which relatedness is less central to intrinsic motivation than autonomy and competence.

Feedback is thought to be important for intrinsic motivation (Deci 1971: 114; Harackiewicz 1979: 1360).[179] According to a meta-study positive feedback enhances intrinsic motivation especially for college students but less for children (Deci, Koestner, & Ryan 1999: 638). It may inform subjects about competence and therefore foster intrinsic motivation (Deci, Koestner, & Ryan 1999: 653). However, negative feedback that foster perceived incompetence tend to undermine intrinsic motivation (Deci & Ryan 2000: 235).

Free choice and the opportunity for self-direction foster intrinsic motivation (Deci, Connell, & Ryan 1989: 580-581). Controlling conditions inhibit autonomy and decrease intrinsic motivation. Albert Einstein (1949: 17) wrote on this topic (see also Einstein's parable of a beast of prey and the undermining of intrinsic motivation in chapter 5.6):[180]

> *This coercion {the need to learn for examinations} had such a deterring effect … that, after I had passed the final examination, I found the consideration of any scientific problems distasteful to me for an entire year. …*

---

[178] For a discussion how autonomy influences organizational citizenship behavior see Cirka (2005).

[179] However, feedback can also be perceived as controlling and decrease intrinsic motivation (Deci, Koestner, & Ryan 1999: 629-630).

[180] That one should not have too much confidence in anecdotal evidence is shown by the counterexample of Pythagoras (Singh & Lynch 1998: 8). Initially, Pythagoras did not succeed to find any student interested in his mathematical reasoning. That is why he paid a young boy three oboli for each lesson he attended. With the time going on, however, the student's initial reluctance to learn was transformed into enthusiasm for mathematical knowledge. In consequence, Pythagoras stopped paying the student who turned finally in an igneous disciple. In this example external regulation increased intrinsic motivation.

It is, in fact, nothing short of a miracle that the modern methods of instructions have not yet entirely strangled the holy curiosity of inquiry; for this delicate little plant, aside form stimulation, stands mainly in need of freedom; without this it goes to wreck and ruin without fail. It is a very grave mistake to think that the enjoyment of seeing and searching can be promoted by means of coercion and a sense of duty.

Intrinsic motivation is considered especially important for creativity (Amabile 1997).[181] For Amabile (1996: 15) intrinsic motivation is the cornerstone of the social psychology of creativity. Raymond argues that it is a "well-known fact that in any kind of creative work motivation is more than half the battle and if our people are nothing else, boy are they motivated!" (Bollinger, Raymond, & Trader 1999: 87). In an experiment Koestner et al. (1984: 242-243) showed that creativity is disrupted by a controlling environment. Rewards in general undermine intrinsic motivation (Deci, Koestner, & Ryan 1999: 647). This holds true for engagement contingent, completion contingent, and performance contingent rewards.[182] There are three notable exceptions. Verbal rewards or positive feedback, unexpected rewards, and task-non-contingent rewards have no negative effect on intrinsic motivation. Analogue to rewards threats (Deci & Cascio 1972, reported in Ryan and Deci 2000), deadlines (Amabile, DeJong, & Lepper 1976: 97), controlling directives (Koestner et al. 1984: 241), and competition pressure (Reeve & Deci 1996: 29) decrease intrinsic motivation.

A very early contribution researching the importance of fun and enjoyment were Csikszentmihalyi's (2000) reflections.[183] He asked why people risk their live climbing rocks, devote their live to art, or spend their energy playing chess (Csikszentmihalyi 2000: 1). Based on interviews he found that individuals who play piano or execute similar tasks experience what he calls flow (Csikszentmihalyi 2000: 35-40). A sign of flow is the merging of action and awareness as well as the centering of attention on a limited stimulus field. A tennis player pays undivided attention to the ball and the opponent and forgets the outside. In such situations the sense of time is often lost (Csikszentmihalyi 2000: 65). In most cases the time passes much faster: an hour pass in a minute (Csikszentmihalyi 1991: 66-67). However, under special circumstances the reverse occurs. Ballet dancers for example report that a difficult turn that takes less than a second in real time stretches out for what seems like minutes. Csikszentmihalyi and Geirland (1996) describe flow in the following way:

Being completely involved in an activity for its own sake. The ego falls away. Time flies. Every action, movement, and thought follows inevitably from the previous one, like playing jazz. Your whole being is involved, and you're using your skills to the utmost.

While the theoretical convergence between Csikszentmihalyi's flow theory and Deci and Ryan's self-determination theory is considerable there are also some divergences (for a discussion for similarities and differences see Deci & Ryan 2000: 260-261). Prominently, Csikszentmihalyi does only consider an optimal challenge but not autonomy, competence, and relatedness as a prerequisite for flow.

---

[181] The *conceptual* definition of creativity of Amabile (1996: 35) is based on two components. *Firstly*, creativity is related to a task that has a heuristic (no cognizable path to a solution) and not an algorithmic (exact solution) solution. *Secondly*, the solution for a problem is novel and appropriate, correct or valuable as well as useful. According to her *consensual* definition something is creative if appropriate observers independently agree that it is creative (Amabile 1996: 33). An observer is appropriate if he is familiar with the domain.

[182] Engagement contingent rewards are based on engagement independent whether a task is completed or performed well. Completion contingent rewards are paid if a task is completed. If the reward is contingent on performance individuals with a better score or result receive a higher reward.

[183] Fun seeking is related to current joy and therefore an affectional action as Weber (1990: 12) called it.

Raymond (2004: 10) argues that an important strength of Unix was not only its technical superiority to other operating systems[184] but also that it "is fun to hack". He argues that the fun factor was in the beginning mainly responsible for the success of Unix. Programmers started to contribute software because it was fun to do so. Ritchie (1978) writes that Unix was never a project "to meet any specific need except that felt by its major author … for a pleasant environment in which to write and use programs". Unix was initially written for the PDP-7 computer because Ken Thompson was familiar with it. He knew it because he wanted to play his favorite computer game called Space Travel and ported it to the PDP-7 (Ritchie 2001). The PDP-7 was hardly used at the Bell Laboratories that allowed playing games on it. By exploring the solar system in the game Space Travel the creators of Unix explored the PDP-7 as a byproduct that in turn influenced Unix development. A similar story is told about Apple's operating system that is claimed to be written by Steve Wozniak to play his favorite game in color (Feller & Fitzgerald 2002: 29).

Fun is certainly not only important in Unix development but also in FOSS development (see for example Lakhani & Wolf 2005). Torvalds (see also Torvalds & Diamond 2001; 1998) argues that most *good* programmers do programming because it is fun. Enthusiasm about what they do is a characteristic of hackers (Himanen 2001: ix). Stallman (2004b) considers fun as the most widespread motive for writing free software. The importance of fun as a motivation seems to be based on different characteristics of FOSS development:

i. Most prominently the high level of ***autonomy*** in FOSS projects facilitates the emergence of intrinsic motivation. Programmers are free to choose the projects and tasks they want to accomplish. Every project member may leave a project whenever he wants. Furthermore, in FOSS development there are hardly task deadlines. For example in the FreeBSD study 96% of developers report that there was no task deadline for their latest source code contribution (Jorgensen 2000, 2001). Therefore the locus of control of FOSS developers is internal respectively behavior is self-determined.

ii. Related to the feeling of ***competence*** is the optimal challenge. Individuals should neither be underchallenged nor overstrained. In the case of FOSS development the optimal challenge is assured by programmers choosing projects and tasks themselves. It may be argued that programmers know themselves best how to apply their skills. Additionally, in the process of peer review feedback is provided. However, feedback in FOSS development may also be harsh and nonproductive. In such a case peer review may influence intrinsic motivation negatively.

iii. Depending on the size and structure of a FLOSS project its members may have a sense of ***relatedness***. The project members may feel to be connected to each other and being part of a community.

Comparing the basic psychological needs autonomy, competence, and relatedness in FOSS development it seems to be clear that autonomy is meet to a higher degree than the two other basic needs. Autonomy of developers is always high independent of size and structure of the FOSS project. A member of a one-man project may feel to the same degree autonomous as a contributor to the Linux kernel. The only exceptions are probably salaried developers. For very small projects the need for competence and relatedness may not be satisfied because peer interaction and peer review require a minimal project size. Additionally, there are contributors that are not connected to other project members. Developers who only contribute one or a few bug fixes are an example. As already mentioned the need for competence may remain unsatisfied if the

[184] In fact, Multics is thought to have been more technically advanced than Unix. However, the problem of Multics seems to have been its huge complexity and the "increasing obviousness of the failure of Multics to deliver promptly any sort of usable system, let alone the panacea envisioned earlier" (Ritchie 1996).

feedback is negative or rough. Indeed, anecdotal evidence suggests that peer review in FOSS projects is not always supportive. It can be concluded that the need of autonomy is usually satisfied in FOSS projects while the satisfaction of needs for relatedness and competence depend on specific project characteristics.

That intrinsic motivation may also be important in commercial settings, especially for creativity, is impressively shown by Google. Google has a “20 percent time” rule that allows the engineers to work on whatever they are interested in (Google 2007). Vice President of Search Product and User Experience Marissa Ann Mayer (2006) argues that one has to take good people, give them the time to work on what they are passionate, and creativity will be the result. In fact, half of the new product launches in 2005 originated from this “20 percent time” rule.[185]

3.2.2 Extrinsic motivation

Extrinsic motivation refers to the performance of an activity in order to attain some separable outcome (Ryan & Deci 2000b: 71). Below integrated and identified regulation will be discussed first followed by introjected and external regulation.

3.2.2.1 Integrated and identified regulation: Pro-social motivation

Integrated regulation occurs if regulations are fully assimilated to the self (Ryan & Deci 2000a: 62). These regulations are brought into congruence with the individuals’ other values and needs. In identified regulation an individual acknowledge the importance of certain behavior and identifies as his or her own. Although integrated and identified regulations are not similar it seems to be difficult to grasp the distinctions empirically and to apply the theoretical differences. In the following, integrated and identified regulations are discussed jointly.

Pro-social behavior is based on the notion that individuals’ utility functions are interdependent (Meier 2004: 17).[186/187] Individuals do not only care about their self-interest but also consider others’ well-being. Pro-social behavior is then based on the individuals’ values and therefore an expression of integrated or identified regulation.

[185] According to Google CEO Erich Schmidt (Battelle & Schmidt 2005) “virtually everything new seems to come from the 20 percent” rule.

[186] It is sometimes argued that pro-social behavior is in fact in the self-interest of the individual. Individuals who help others may get help in the future themselves. Pro-social behavior is then in reality a tit for tat strategy. Other people help because of the related reputation of being a kind-hearted person. Yet other people help because of the fear of being sanctioned if they do not. Such arguments may be good explanations for many pretended pro-social behavior. In fact, one should not claim without proper theoretical reasoning that behavior is altruistic. Nevertheless, there is also real altruistic behavior in anonymity where strategic interaction and reputation effects can be excluded. North (1997: 20) comes to conclusion that “human behavior appears to be more complex than that embodied in the individual utility function of economists’ models” and “that many cases are ones not simply of wealth-maximizing behavior, but of altruism …”. Arguments that altruistic behavior is in the self-interest because it produces some sort of inner satisfaction may be logically sound but seem to be tautological and therefore useless for further theoretical advancement.

[187] According to Meier (2004: 17-20) there are three prominent formulations of pro-social behavior. The utility of other individuals can influence i) the own utility directly (pure altruism), ii) the own utility because helping others produces a warm glow (impure altruism), or iii) the own utility depends on the difference between one’s own and another’s well-being (inequality aversion). However, this difference is to fine-graded for the following discussion and consequently neglected.

Having access to computer resources is very central to hacker ethic (Himanen 2001: x; Levy 1984: 40). Contributions to FOSS projects may then be based on an inner obligation to make or keep software free. The most obvious example of a programmer adhering to such an inner obligation is Richard Stallman, a man who called himself back in the eighties the last true hacker (Levy 1984: 413). In his project announcement of the GNU project Richard Stallman (1983) writes:

> *Why I Must Write GNU*
>
> *I consider that the golden rule requires that if I like a program I must share it with other people who like it. I cannot in good conscience sign a nondisclosure agreement or a software license agreement. So that I can continue to use computers without violating my principles, I have decided to put together a sufficient body of free software so that I will be able to get along without any software that is not free.*

For Richard Stallman and the FSF free software is a matter of liberty and freedom (FSF 2006d: see also the definition of free software). However, this does not exclude selling free software. As the FSF argues free software has nothing to do with price. One should think about free speech and not about free beer. In consequence, the GNU GPL does not regulate whether one charges something or not for distributing free software. One may charge nothing, a penny, a dollar, or a billion dollars.

For adherents of free software the freedom of software is the very first criterion to judge software. In consequence, they prefer using inferior free software to superior non-free software. Stallman (1999) writes on non-free books from O'Reilly:

> *If they do an important job, that is not a reason to use them - it is a reason to replace them, one way or another, with free substitutes.*

In the FLOSS-US study 47% of developers strongly agreed that a reason to participate in FLOSS development is that everybody should be free to modify the software they use (David, Waterman, & Arora 2003b: 19). This was, apart from the category "another reason"[188], the most strongly agreed motivation.

Such a pro-social behavior is in its ideal form what Weber (1990: 12) called "wertrational" (value-rational).[189] Pro-social motivated FOSS hackers contribute to a project because it is a valued goal. They are therefore influenced by value-rational behavior.

Although sometimes fun and pro-social motivation is mixed and thought to be connected to each other this must not be necessarily the case. One may have fun writing FOSS software without being interested whether it is FOSS or not. For Torvalds (2001: 165) having fun is something distinct from idealistic motivations. As Stallman (2004b) puts it: "The determination to live in freedom is an important motive for many, *but not all*, free software developers" {emphasis added}.

---

[188] This unusually high fraction developer choosing the category "another reason" indicates that the list of reasons why developers have joined FOSS development is not exhausting. Indeed, the motivation fun seeking as an answer is almost completely missing. That the missing of fun seeking categories may be a major reason for the very high fraction of "another reason" is backed by a superficial own analysis of the raw data (David, Waterman, & Arora 2003a). Many developers responded to the open item about their motivation to participate with "just for fun" or similarly.

[189] However, most FOSS hackers in reality probably would not do so if success were nil. Value-rational behavior is then accompanied by goal-instrumental behavior (zweckrational). Such developers would only contribute if i) the goal were valued and if there were ii) at least a dwarfish possibility for success. As Weber (1990: 13) suggests social action in reality is mostly a mix different ideal types.

3.2.2.2 Introjected Regulation: Peer recognition

Introjected regulation is based on guilt and anxiety avoidance or attaining of pride and ego-enhancement. An individual acts to enhance or maintain self-esteem and the feeling of worth. Individuals are concerned about their reputation within the community they live (Akerlof 1980: 753). Approval from others respectively *peer recognition* is therefore an important motivational drive.

Polanyi (2000: 5) argues that scientists are primarily motivated by the merits provided by the scientific community. Robert Merton (1979: viii) considers the citation index[190] as a currency in the academia as well as a kind of registration of intellectual property. He writes that citations "are designed to repay intellectual debts in the only form in which this can be done: through open acknowledgement of them". Merton argues that citations as a currency have the paradoxical character that the more widely scientists make their intellectual property available to others the more securely it becomes identified as their property. Because science is widely based on recognition by peers the only way to be repaid for the own research is publication (Kelly 2005: 421; Merton 1979: viii). Similar to science peer recognition is an important aspect of motivation in FOSS development (Bezroukov 1999a; Himanen 2001: 6; Raymond 1998). For Bezroukov (1999a) programming is a special kind or at least a close relative of scientific activities. He lists many similarities such as nil duplication costs of results and easiness of minor modification. Participants are in both cases mostly not solely motivated by money.

Linus Torvalds (2001: xx-xxi) reports that Linux development is comparable to global team sport and that the social part of Linux is very important. FOSS developers are motivated by the esteem they gain in the eyes of their peers (Torvalds & Diamond 2001: 122). Raymond (1999a: 62) observes that while coding remains an essentially solitary activity the great hacks harness the attention and brainpower of an entire community. Collective identification is an important aspect of the FOSS community (Bonaccorsi & Rossi 2003: 9). In consequence, Torvalds' (2001: 87) decision about the timing of the first release of Linux[191] depended on the question "when is it good enough that I won't have to be ashamed of it?". Bruce Perens, a known member of the FOSS community, notes that "programmers are like artists" and that they like to show their best stuff for their peers (Hof 2003: 86). To do so is possible in FOSS development because the source code is open respectively free while it is hidden at most corporations.

Peer recognition is deeply rooted in the hacker ethic (Himanen 2001: ix-x). The importance of peer recognition is corroborated by empirical results. In the FLOSS-US study 24% strongly agree that a reason to participate in FOSS development was to interact with like-minded programmers (David, Waterman, & Arora 2003b: 19).

Related to peer recognition is peer pressure. Personal habits or individual believes are overridden to conform towards group norms. Unix pioneer Doug McIlroy (McIlroy 1991; cited in Raymond 2004: 32) observes that peer pressure and simple pride in workmanship is very important in software development.

Extrinsically motivated individuals are not inherently interested in the activity itself and they must be externally prompted (Ryan & Deci 2000a: 64). However, in some cases it may be of value that an individual is intrinsically motivated. If so, a process of internalization must start in that extrinsic motivation is turned into intrinsic motivation. An important way to start internalization is the family, a peer group, or a society. Such

[190] For an in-depth analysis of citation indexing see Garfield (1979).
[191] At this time Linux was called Freax (Torvalds & Diamond 2001: 88).

groups are often the groundwork for facilitating internalization because there is a human need for relatedness. Therefore, peer recognition is not only an important extrinsic motivation itself but may also be a trigger for intrinsic motivation. A software developer may be motivated by other peers to participate in FOSS development but with the time going on he starts being interest in the programming itself.

3.2.2.3 External Regulation: Paid development, personal requirement, investment in reputation, and investment in human capital

An externally regulated individual performs action to satisfy an external demand or to get an externally imposed reward. For Lerner and Tirole (2005: 56-58) developers will contribute to a FOSS project as long as the benefits exceed the costs. The net benefit of contribution consists of immediate (current benefit minus current cost) and delayed payoffs (delayed benefit minus delayed cost). The costs of FOSS participation are mainly opportunity costs. Immediate benefits are *being paid* for participation and *personal requirements* for software. Delayed benefits include *career promotion* based on job market signaling and *improved programming skills* (human capital). These benefits will be discussed in turn below.

Paid development / business requirements

In some FOSS projects there are paid developers (see chapter 2.4). The puzzle in this case is not why these individuals are motivated. Salary as a mean to motivate employees is established and has therefore not to be explained further (Baker, Jensen, & Murphy 1988; Furnham & Argyle 1998; Lawler 2000; Mitchell & Mickel 1999: 570-571). Rather puzzling is *why* they are paid. Doing business is not a principle contrast to FOSS. The FSF “is not against business” and ensure its activities itself with doing business (Stallman 2001a). However, whatever the business model it must acknowledge the terms of FOSS licenses. Therefore, commercial firms do not get exclusive rights to exploit their contributions. It may even be that the contribution to a FOSS project under a BSD license is appropriated by a competitor. However, it is usually thought that the protection of intellectual property fosters innovation because inventors receive exclusive rights to exploit their products for a certain time. Nevertheless, it can be assumed that commercial firms have some sort of incentive to pay FOSS developers. In the following, these incentives respectively the reasons for firms to participate in FOSS development are discussed.

There is a short history of literature on firms or individuals who freely reveal their innovations. There are examples from iron and steel production (Allen 1983), sporting goods (Franke & Shah 2003), and library information systems (Morrison, Roberts, & von Hippel 2000). The explanation for this behavior is the hope of support by a user community (Franke & Shah 2003) or by other manufacturers (Harhoff, Henkel, & von Hippel 2003). Building up a reputation as a good community member may also be a reason why commercial firms reveal innovation.

Several authors discussed why commercial firms should contribute to FOSS projects. Behlendorf (1999: 161) argues that contributing to an existing FOSS project enables to take advantage of its momentum. This leads to higher code quality code, marketing lead generation, and common platform establishment. Hecker (1999: 46) concludes that a company can reveal its contribution freely while stile serving its own business interests. He specifies the following business models (Hecker 1999: 49):[192]

- ***Support Seller:*** Revenues come from training, consulting, custom development, and post-sales support.
- ***Loss leader:*** A FOSS project is used as a loss leader for commercial software. In marketing a loss leader is a pricing strategy where something is given away below cost to stimulate profitable sales of other goods.
- ***Widget frosting:*** FOSS software such as drivers is needed to sell hardware.
- ***Accessorizing:*** Physical items supportive of FOOS such as books are sold.
- ***Service Enabler:*** FOSS is created to support access to online services.
- ***Brand Licensing:*** Other companies are charged for the right to use its brand names and trademarks for creating derivative products.

In case a commercial firm owns the complete copyright a dual license strategy is also possible. Prominent examples are MySQL and Trolltech. It seems *subjectively* that the financial success of these firms is heavily due to their proprietary licenses while the success of the product is heavily due their FOSS license.

Henkel (2006: 955) identified four reasons why commercial firms contribute to FOSS projects contribute and three reasons why they do not:

Reasons to contribute

1. Standard setting and enabling compatibilities
2. Increasing demand for complementary goods and services
3. Benefiting from external development
4. Signaling technical excellence or good citizenship in the FOSS community

Reasons note to contribute

1. Freely available software cannot be sold anymore
2. Software can be used by competitors which may imply a loss of competitive advantage
3. Loose of control over software development

Figure 40: Reasons for commercial firms to contribute or not to contribute to FOSS projects

In a quantitative study Henkel (2006: 966) finds that in the field of embedded Linux commercial firms contribute about half of the code they have developed for FOSS projects. Firms reveal source code *firstly* because the GNU GPL requires it, *secondly* because they want to appear as a good player in the community, and *thirdly* to get bugs fixed (Henkel 2006: 961). A further reason is that being included in a standard distribution reduces maintenance efforts and probably other developers add new functionalities. Not important for revealing source code is to set a standard or to identify potential employees. Generally, firms with higher experience with the FOSS development process reveal more of their source code (Henkel 2006: 967). To sum

[192] Hecker (1999: 49) specifies two additional models. However, one is a combination of other models while for the other it is not comprehensible how revealing code generates somehow revenues.

up, besides motivation based on legal necessity the reasons for firms to reveal source code is firstly signaling good citizenship in the FOSS community and secondly benefiting from external development.

In an interesting study Bonaccorsi and Rossi (2004) compare the motivation of firms and individual programmers to contribute to FOSS projects. Commercial firms are primarily driven by economic motivation (e.g. gain reputation among customers and competitors), secondly by technological motivation (e.g. feedback of community to improve the software) and thirdly by social-based motivation (e.g. agreement with FOSS values). In a comparison with different studies on individual motivation Bonaccorsi and Rossi find that individual developers attach more importance to social-based motivation. While firms do not participate in FOSS projects to enhance human capital this is a major reason to contribute for individual developers.

It may be that paying developers for FOSS participation change their motivation [Frey, 2001 #1702; Deci, 1999 #1496, see also chapter 5.6]. The Linux community for example feared that the employment of Linus Torvalds by the commercial firm Transmeta might change his attitude towards Linux. However, it finally turned out that it was not his association to Transmeta that changed his involvement in the development process of Linux but the birth of his first daughter (Moody 1997).

Personal requirements

For von Hippel (2005a: 19-22) an important part of innovation is based on user activities. Such users share mostly the characteristics of lead users (von Hippel 2005a: 22). Lead users are defined as members of a user population who:

i. experience a need far earlier than other users and
ii. highly benefit from obtaining a solution adjusted to their need.

A professional windsurfer for example has special needs concerning equipment that a regular windsurfer has not yet. He benefits if he helps innovating better equipment. With the time going on regular windsurfers will request the improvements as well. Similarly, an “almost right” basketball shoe may be acceptable for a weekend athlete (Franke & von Hippel 2003: 1199). Therefore, he does not benefit enough from an improvement and does not invest scare resources such as time for improvement. A professional player on the other hand benefits even from small improvements and may invest a considerable amount of time to help a manufacturer to produce a new basketball shoe.

In the FLOSS-US study 31% of developers strongly agreed that a reason to participate in FOSS development is that they needed an additional feature and 14% strongly agreed that bug fixing was another reason (David, Waterman, & Arora 2003b: 19). It may be assumed that a considerable amount of these developers share to some extent the characteristics of a lead user. A concrete example of lead users is webmasters that contribute modifications and amend features to the Apache HTTP server to satisfy their needs (Franke & von Hippel 2003). Almost one fifth of the Apache users installed a customized version of the software that made coding necessary (Franke & von Hippel 2003: 1209).

According to Raymond (1999a: 32) FOSS projects usually start by a developer scratching a personal itch (see also Feller & Fitzgerald 2002: 139-140). Scratching an itch is very often cited to be motivation for FOSS development. A user has a personal requirement and solves the corresponding problem by implementing a solution in a new or in an established FOSS project. In the following scratching an itch is discussed in length.

Scratching an itch may or may not be related to the arguments made for lead users. Developers who scratch an itch have most probably needs earlier than the regular computer user. However, lead users are purely motivated by a rational cost-benefit calculation in which opportunity costs must be considered. Starting a new FOSS project or contributing to an existing project is then only rational if the cost - benefit of the next best solution is lower. Such a solution may be "buying new hardware", "buying new software", "accept the bug", or "accept inadequate functionality".

It must be questioned whether developers always benefit from meeting a personal requirement by scratching an itch in a strictly economic sense. Buying new software and hardware is in many cases not very expensive. Additionally, revealing source code to others is costly. Writing source code that works properly is one thing. Source code that can be maintained by the developer itself is more troublesome. Additionally, source code that works properly for other unknown users increases workload again. Finally, writing source code that can be maintained by unknown users increases the workload considerably. Therefore, releasing usable and maintainable source code to the outside is not costless as sometimes suggested.

If a developer has a software problem and solves it this must not necessarily be done because the problem must be solved imperatively. One may be challenged solving a problem or one is curious to know why something does not work. In the former case the motivation is tightly related to fun and in the later case related to learning. Based on this argumentation scratching an itch is in the end not always based on a personal requirement. In these cases scratching an itch is a trigger for software development but not the main motivation.

An example of a developer for whom scratching an itch was *probably* the trigger for FOSS involvement but not the real motivation is Roger Luethi. He is a Linux maintainer responsible for the VIA Rhine network card driver. Luethi had such a network card and wanted to run Linux. However, the driver did not work properly. In consequence, he started to fix and clean the driver in the Linux kernel. Clearly, an itch had to be scratched. Nevertheless, it seems to be very questionable how Luethi should have profited by fixing the driver in a hard-nose economical view. *Firstly*, the Via Rhine network card is a cheap low-end device. An obvious alternative to fix the driver problem had been to invest 30$ and buy a more advanced network card which is known to work properly under Linux. Instead of fixing the problem he could have invested the same amount of time in working as a paid programmer or the like. The corresponding earning would have exceeded the cost of a network card within a very short time. *Secondly*, by referring to personal requirement one may not explain why he did not stop contributing to the Linux kernel after his problem was solved. In fact, he started to fix other people's problems. Therefore, it may be concluded that the decision to fix his initial problem with the network driver would have been economically a very bad choice. The problems with the driver were a trigger to start contributing. The real motivation could have been the promotion of FOSS, learning, or to have fun being challenged.

A similar example against equating scratching an itch with personal requirement is the occurrence Richard Stallman mentions being the trigger for the GNU project (see fore example Lerner & Tirole 2005: 61). Richard Stallman wanted to fix the problems with a new laser printer that jammed frequently at the Artificial Intelligence Laboratory at MIT (Williams 2002: 1-12). However, he did not get access to the source code of the printer that made its repair difficult. Without access to the source code a software problem cannot be fixed directly. Because of this it is a good thing to have access to the source code. Due to his annoyance of not having access to the source code Stallman finally quit his job and started the GNU project. He had the following goals (Stallman 1983):

To begin with, GNU will be a kernel plus all the utilities needed to write and run C programs: editor, shell, C compiler, linker, assembler, and a few other things. After this we will add a text formatter, a YACC, an Empire game, a spreadsheet, and hundreds of other things. We hope to supply, eventually, everything useful that normally comes with a Unix system, and anything else useful, including on-line and hardcopy documentation.

With this statement Stallman intended to fulfill an extremely ambitious and extremely time-consuming project. Although having an operating system with access to the source code does have benefits, the costs (particularly opportunity costs) for writing it are tremendously high. Byte, which was perhaps the most influential personal computer magazine of the seventies and eighties, called the GNU project "probably the most ambitious free software development project to date" (Betz & Edwards 1986).

The printer jam respectively a personal requirement was a trigger for Stallman's endeavor but not the real motivation. For Stallman the real reason for starting the GNU project and bear the related costs was that he felt an inner obligation to do so. He did not wanted to use software which are "violating {his} principles" (Stallman 1983) and cannot be shared "with other people who like it". He would even use free software if it had no technical advantage over a proprietary system (Stallman 2001a).

Yet another example of scratching an itch that is not or not solely related to personal requirement is Donald Knuth's involvement in developing the typesetting system TeX (Vose, William, & Knuth 1986). The reason to start TeX was to have a convenient tool to write his famous multivolume "The art of computer programming" (Knuth 1997). However, after some time he started having fun in the development itself. Knuth (2002: 321-322) also mentions a downside of having programmed TeX:

The downside is that I'm too sensitive to things now. I can't go to a restaurant and order food because I keep looking at the fonts on the menu. Five minutes later I realize that it's also talking about food.

That Knuth is still interested in typesetting although his personal problems with typesetting are solved clearly shows that his motivation for scratching the itch was probably beyond a personal requirement.

As discussed in length above scratching an itch may be only the trigger but not the real motivation to contribute. In such cases pro-social orientation, learning, or fun is the real motivation. This does not mean that lead user innovation does not happen. However, it seems to be that in some cases scratching an itch is wrongly related to a personal requirement.

Investment in reputation

In most job markets the employer may not be sure about the productive capabilities of an individual at the time he hires him (Spence 1973: 356). Hiring an individual is then an investment decision under uncertainty. However, the employer may infer the productivity indirectly from personal data in the form of observable characteristics and attributes such previous work, education, sex, or criminal records (Spence 1973: 357). Some attributes and characteristics such as previous work or education can be manipulated by the individual while others such as sex can hardly be changed. By manipulating attributes and characteristics valuable signals can be produced. Whether an individual produces such signals depends on the related costs and benefits. It is assumed that the costs of producing a signal are negatively correlated with the productivity of an individual (Spence 1973: 358). Therefore, an employer can assume that a superior signal is likely based on superior productive capabilities.

Delayed benefits from contributions to FOSS projects include career promotion based on job market signaling respectively reputation. Some authors suggest that such signaling respectively reputation is in fact a motivation of FOSS developers (Egyedi & Wendel de Joode 2003: 91; Lee, Moisa, & Weiss 2004; Leppämäki & Mustonen 2004; Lerner & Tirole 2005; Lerner, Tirole, & National Bureau of Economic Research. 2000). By signaling their capabilities the developers hope to get a better job in the future. Contributions to a FOSS project are in such a case a career investment. Such an investment assumes that a human resource manager makes the following conclusion (Lee, Moisa, & Weiss 2004: 5):

> *Candidate A has drafted vital modules for the Linux program. Linux is a prominent brand. (This is the reason why I would know of it.) Since the product is widely known and used (especially by software experts), it must be good. Apparently, Linux has high quality requirements and a lot of good programmers involved. (Otherwise, the product would not be so successful.) So, if this guy has made it into the changelog file, he must be very skilled. We should offer him an adequate salary.*

There are two possible categories of signals in FOSS development:

i. In most FOSS projects the names of the contributors are stored in credit lists or listed on a project website.[193] In the KDE project 518 contributors are for example listed by name on the project website (KDE 2007). From this list one may learn that Aaron Granick contributed to "KIRC" while Aaron Seigo contributed to "a variety of KDE libs and apps".[194] One may argue that commercial firms have no incentive to display the achievements of their best programmers because this raises the possibility that the concurrence poach them. Such a fear does hardly apply to FOSS development.

ii. It is thought that firms looking for a certain skill in the labor market can do so by browsing source code and identifying thereby qualified software developers (Lerner & Tirole 2002: 216, 2005: 60). By doing so one may evaluate the work very accurately. Indeed, Torvalds (2001: 122) notes that very prolific FOSS developers attract the attention of employee who searches the source code to spot and hire top programmers. This is only possible because the source code is accessible (Lee, Moisa, & Weiss 2004: 4).

The production of some signals is more profitable than others. Economic theory suggests that signaling incentives are stronger (Holmstrom 1999; Lerner & Tirole 2002: 214):

i. the more visible performance to the relevant audience is

ii. the higher the impact of effort on performance is

iii. the more informative the performance about talent is

Lee et al. (2004: 4-5) suggest that the *visibility* and *credibility* of a signal rise with the number of programmers contributing to a project. *Firstly*, developers are users themselves and thereby rise the number of users directly. *Secondly*, the number of developers augments the product quality that in turn will raise the number of users in the future. *Thirdly*, developers are technological trendsetters indicating future success. Franck and Jungwirth (2003a: 12) argue similarly that the value of a signal increases with project size because big projects i) attract more attention and ii) increase the competition between the developers that in turn makes it more difficult to

[193] It is a taboo to remove a person's name from a project history, a credits list, or a maintainer list (Feller & Fitzgerald 2002: 96; Raymond 1998). This norm could be explained by the intent to protect valuable signals.

[194] These names have been selected because they are first and second in the list.

get code merged. Therefore, a signal produced in a FOSS project with a big developer community is more valuable. However, if more programmers are involved in a project the competition for a limited number of signals is higher. In consequence the higher the number of programmers contributing to a project the lesser the likelihood that an individual can produce a signal (Franck & Jungwirth 2003a: 14).

Although signaling effects are often cited as a source of motivation for FOSS development it is not easy to understand how it works in practice:

i. Information in credit files or on project websites do not contain information about performance or about the difficulty of the task. Mostly there is only the information that somebody contributed to a task. Such minimal information is also included in job references. So it may be argued that the mere listing of contributors is in fact nothing else than an inferior variant of job references. The only two advantages of credit files and credit websites are that the information is accessible a) online and b) without having to request it actively. However, such information can sometimes also be found for commercial software. If one looks at the About dialogue of the Adobe Acrobat 6 software one may see the names of the developers as well.

ii. There are two problems associated with browsing source code to find an employee. *Firstly*, an employer doing so has to understand whether the written source code is ugly or awesome, whether the task was hard or not, or whether the solution is clever. To make such a judgment about a specific part of the Linux kernel may be easy for an involved peer or an experienced hacker. However, regular software programmers and human resource departments most probably have a very hard time to understand the source code. There may be interaction with other parts of the source code that is not obvious to an outsider or some strange source code may be a work around due to hardware errors. *Secondly*, browsing source code is time consuming and therefore not free.

iii. One aspect that influences the value of signals is its exclusiveness. A signal that is produced by everyone is worth nothing while a signal singular in the world may be very valuable. Therefore, only a fraction of FOSS developers respectively the top developers can rationally anticipate producing a valuable signal that exceeds its costs. A rational individual motivated by signaling his skills should take the likelihood for producing the signal into account (Hertel, Niedner, & Herrmann 2003: 1163). Unfortunately, many papers that argue that reputation is an important motivation in FOSS development take for sure that an individual who intends to produce a valuable signals succeeds.

iv. The few developers who have benefited from reputation effects economically started contribution at a time in which they were considered being geeks and contribution to FOSS development was a strange thing. Nobody would have ever thought that Linux might challenge Microsoft Windows or even Unix. There are developers who profit economically now from their reputation but back decades ago they could have hardly been motivated by producing a valuable signal.

In an item related to the motivation of commercial firms revealing source code in a FOSS project the *least* agreed answer was to identify potential employees by looking at the contributions to the source code (Henkel 2006: 961). Based on this observation it may be concluded that even if developers seek to gain reputation in order to get a job firms do not put much value to such signals.

It can be inferred that theoretically reputation effects may be a source of motivation to contribute to FOSS development. As a former project leader of the Debian community explains: "Reputation does help: when yon see a good name, you will take a look" (Egyedi & Wendel de Joode 2003: 91). However, practically it seems to be that reputation effects are overrated in their relative importance by the scientific community.

Investment in human capital

Activities that influence future real income through the imbedding of resources in people are investments in human capital (Becker 1962: 9). Such activities may for example include schooling, on the job training, or the acquiring of information about the economic system.

Software developers learn new skills and perfect old one while solving coding problems. Thereby future productivity is improved at the cost of scare time resources. A developer receives lower total compensation during the time he contributes to a FOSS project but later his earnings will perhaps increase. Depending on the projects and the tasks a developer accomplish the learned skills are a mix between general and specific new abilities. For some developers contributing to FOSS projects is therefore an investment in human capital. In the following it is discussed i) whether one may really learn something from participation in FOSS projects, ii) what is learned, iii) what is learned better in FOSS projects, and iv) how learning works.

Is something learned?

In the FLOSS-Asia study 86.5% of subjects reported that they have learned very much from participation in FOSS development (Mitsubishi Research Institute 2004a). Only 13.5% report that they have learned only little and nobody reported that he has learned nothing. In the FLOSS-US study 37% strongly agreed that a motivation to participate in FOSS development was to become a better programmer (David, Waterman, & Arora 2003b: 19). In the FreeBSD study 90% of the developers report that the knowing that their source code may be read by other competent developers encouraged them to improve their skills (Jorgensen 2000, 2001).

What is learned?

In the FLOSSPOLS project special attention was given to learning (Ghosh & Rüdiger 2006). It was asked how much FOSS project members have learned for an array of different skills (see Figure 41). Interestingly, FOSS developers seemed to have learned most about legal issues such as the difference between copyrights, patents, and licenses.

Exactly half of the subjects state that they have learned to reuse code written by others. Further important skills learned are to run and maintain complex software systems, basic and introductory programming skills, to look for and fix bugs, to become familiar with different programming languages, and how to write code in a way so that it can be reused.

Developers learned two general skills by contributing to FOSS projects. Subjects state that they learned English and that they got a good overview about the developments in software technology.

	Skills	I learned... a lot	some	little	nothing
Technical	To reuse code written by others	50.0%	34.5%	12.6%	2.9%
	To run and maintain complex software systems	49.3%	29.5%	14.0%	7.2%
	Basic / introductory programming skills	46.6%	24.9%	14.7%	13.8%
	To look for and fix bugs	43.6%	33.4%	18.6%	4.4%
	To become familiar with different programming languages	43.0%	33.2%	20.3%	3.4%
	To write code in a way that it can be reused	42.4%	36.0%	12.7%	8.9%
	To design modular code	32.7%	35.6%	21.6%	10.2%
	To document code	18.2%	42.8%	29.3%	9.7%
	To create new algorithms	11.7%	28.2%	38.1%	21.9%
Managerial	To clearly articulate an argument	27.9%	38.1%	24.0%	10.0%
	To express personal opinions	24.9%	39.9%	25.5%	9.7%
	To accept and respond to criticism from others	24.9%	47.3%	20.7%	7.2%
	To coordinate own work with the work of others	22.8%	43.7%	20.1%	13.5%
	To evaluate the work of others	17.4%	46.0%	25.4%	11.2%
	To lead a project or a group of people	16.9%	27.1%	28.9%	27.1%
	To keep a community going	15.9%	30.0%	33.0%	21.1%
	To clearly define and achieve targets	12.8%	34.6%	29.9%	22.7%
	To settle conflicts within a group	12.6%	36.3%	29.8%	21.2%
	To motivate people	11.9%	27.7%	37.5%	22.9%
	To plan work and stick to a work schedule	5.4%	24.7%	33.4%	36.4%
Legal	To understand the diff. between copyrights, patents, and licenses	57.7%	27.4%	12.9%	2.0%
	To understand licenses	52.0%	37.1%	8.3%	2.6%
	To understand copyright law issues	49.6%	33.6%	14.0%	2.8%
	To understand patent law issues	42.2%	35.0%	18.8%	4.0%
	To improve my understanding of liability issues	24.6%	39.2%	28.9%	7.3%
General	To better understand English, especially technical issues	42.3%	28.9%	12.2%	16.7%
	To get an overview of developments in software technology	40.4%	41.0%	17.8%	0.9%
	To get an overview of needed skills in the software professions	28.9%	42.4%	22.8%	5.8%
	To understand and work with people from different cultures	23.0%	43.1%	24.8%	9.1%
	To interact with other people	20.2%	45.7%	26.1%	7.9%

N=361

Figure 41: Skill improvement because of FOSS involvement[195]

What is learned better in FOSS projects?

As discussed above participation in FOSS development improves certain skills. However, this observation does not exclude that other methods of learning such as taking a formal course are even more efficient.

Almost 71% of developers in the FLOSS-Asia study note that they learn more or much more efficiently by participation in FOSS development than by formalized ways to learn in universities and companies (Mitsubishi Research Institute 2004a). Only 5% of developers think that formalized ways to learn are more efficient.

195 Ghosh et al. (2006: 21). Skills that more than 50% of the subject had learned are marked dark-gray and skills with more than 40% are marked gray.

Certainly, this result is highly biased because no non-FOSS developers have participated in the survey Therefore no conclusion should be drawn.

More reliable and interesting are the results that refer to individual skills. In the FLOSSPOLS study it was asked whether certain skills are better learned by FOSS participation or by a formal computer science course (Ghosh & Rüdiger 2006: 22). The skill to write source code so that it can reused is clearly better learned in FOSS projects. This also applies to accept and respond to criticism, to coordinate own work with others, and to develop an awareness of legal issues relating to software. Clearly better learned in formal computer science courses are basic and introductory programming skills, the clear definition and achievement of targets, and to work and stick to schedule. If these results are evaluated subjectively the results seem to be very sound. For example code reuse is especially characteristic for FOSS development while setting targets is clearly not characteristic.

Skills	**Better learned**	
	in FOSS projects	in formal computer science courses
To write code in a way that it can be reused	80.3%	19.7%
To accept and respond to criticism from others	75.8%	24.2%
To coordinate own work with the work of others	72.8%	27.2%
To develop an awarenenss of legal issues relating to software	72.2%	27.8%
To run and maintain complex software systems	65.7%	34.3%
To evaluate the work of others	64.9%	35.1%
To express personal opinions	56.2%	43.8%
To document code	51.4%	48.6%
To lead a project or a group of people	49.7%	50.3%
Basic / introductionary programming skills	28.4%	71.6%
To clearly define and achieve targets	25.0%	75.0%
To work and stick to a work schedule	13.8%	86.2%

N=361

Figure 42: Skills learned: Participation in FOSS projects vs. formal computer science courses[196]

How one learns

There are different arguments why participation in FOSS projects is better for enhancing programming skills respectively human capital than taking a formal programming course:

i. Many interesting problems emerge only if software is used for production by independent users. It is a different thing to write an operating system for a given and constant set of hardware and requirements

[196] Ghosh et al. (2006: 22).

than doing so for a wide and largely unknown audience with different hardware and different requirements. Users may also use the software in a different way than assumed. By having users FOSS development introduces a strong reality check almost completely missing in formal programming courses.

ii. The probably most important reason why FOSS development enhances programming skills is peer review. It is clear that this only applies to projects above a certain size with a proper peer review. There are two rationales why peer review induces learning:

Firstly, developers learn from general discussions. Following discussion on pros and cons of a special solution may be very helpful for understanding software development. Active participation in the discussion may boost this effect. In the FLOSS-Asia study 11% of the subjects state that they learn software development by participating in FLOSS discussions (Mitsubishi Research Institute 2004a). In the FLOSSPOLS 41.6% of developers state that participating in discussions is a very useful way of learning technical skills (Ghosh & Rüdiger 2006: 24). In the FreeBSD study 93% of developers report that they improved their technical skills by communicating with peers (Jorgensen 2000, 2001).

Secondly, specific comments on submitted source code may be a very helpful feedback that in turn makes learning possible (Bonaccorsi & Rossi 2003: 11; von Krogh, Spaeth, & Lakhani 2003: 1218). In scientific writings that discuss learning effects in FOSS development almost only this argument is examined.

Surprisingly, in the FLOSS-Asia study only 3% of subjects state that they have learned from feedback of the FOSS community to patches they have written (Mitsubishi Research Institute 2004a). Because nothing is known about the project size in the FLOSS-Asia it is not clear whether small project size and a non-existing peer review is the reason for the low importance of feedback on submitted patches or whether the theoretical considerations are wrong. It also may be that the found results are due to Asian culture.

In the FLOSSPOLS study 54.6% of developers state that reading feedback to the own contributions is very useful (Ghosh & Rüdiger 2006: 24). Although this looks impressive it is only the fourth most important way of learning technical skill.

iii. A possibility for raising programming skills is reading and analyzing the source code of other programmers. A software developer who intends to write another MP3 player may browse the source code of other projects and increase his knowledge by doing so. This is only possible because the source code is accessible.

Kernighan and Plauger (1976: book cover) write: “good programming is not learned form generalities, but by seeing how significant programs can be made clean, easy to read, easy to maintain and modify, human-engineered, efficient, and reliable, by the application of common sense and good programming practices. Careful study and imitation of good programs leads to better writing”.

That reading and analyzing the source code of other programmers is important for learning is backed up by an item in the FLOSS-US study. Roughly 23% of the developers strongly agreed that a reason to start contributing is to learn more about how a particular program works (David, Waterman, & Arora 2003b: 19). It is very surprisingly that 41% of developers in the FLOSS-ASIA study state that reading the source code of other developers is the most important way of learning software development skills

(Mitsubishi Research Institute 2004a). In the FLOSSPOLS study 56% of developers state that reading the source code of other developers' is a very useful learning method (Ghosh & Rüdiger 2006: 24). This is thereby the second most important way of learning.

iv. An often underestimated characteristic of FOSS development is code reuse. By doing so a developer may focus on the specific theme he is interested in. To implement a special idea for solving a certain problem one does not have to write the whole software. By reusing source code a developer may concentrate only on the problem he intends to solve. A hacker with an idea for a new desktop environment does not need to write a kernel from scratch to run the desktop environment but can use simply an existing one. This makes learning very effective.

v. By contributing to FOSS development the attention is turned away from theoretical concepts to active writing of source code. This learning by doing approach is a contrast to other educational methods. Torvalds makes the point that if he had a MS-DOS operating system he "wouldn't have learned a thing" (Moody 1997). The saying that the proof of the concept is eating the pudding may also be applied to software development. A software design may be theoretically sound but may fail in productive use.

Contrary to most reflections on human capital enhancement in FOSS development a reservation must be made. It should be noted that the arguments one and two mentioned above do not apply to all FOSS projects. There is a respectable fraction of FOSS projects that have not enough members to ensure the proper functioning of a peer review system. It is also clear that there are FOSS projects that have no other users than the developers themselves.

Concerning the mechanisms of learning it can be summarized that reading the source code of other programmers, following general discussions, and the focus on active writing of source code are probably underestimated in scientific writing while getting comments on contributed source code is probably overestimated.

3.2.3 Recapitulation of the motivations to contribute to FOSS projects

In an interview Stallman (2004b) mentions fun, the determination to live in freedom, being admired, gaining a professional reputation as a capable programmer, gratitude to the community, hatred for Microsoft, and money as possible motives why developers contribute to FOSS development. This listing is comparable to the motivations assumed to be important in this study. However, the hatred for Microsoft is subtracted from this list. It can certainly not be ruled out that to bear hatred towards Microsoft is a motivation for some hackers. Within the FOSS community Microsoft is probably not the most estimated company. Yet, it is thought that the antipathy against Microsoft is not a major motivation.

Personal requirements and investment in human capital are thought to be important types of motivation. These types are therefore added to Stallman's list. In consequence, there are the following generic types of motivations to contribute to FOSS:

- Intrinsic motivation
    - **Fun seeking**
- Extrinsic motivation

- o Integrated and identified regulation
  - **Pro-social motivation**
- o Introjected Regulation
  - **Peer recognition**
- o External Regulation
  - **Paid development / Business requirements**
  - **Personal requirement**
  - **Investment in reputation**
  - **Investment in human capital**

3.3 Motivation in FOSS: An empirical overview

In the following, the results of different empirical studies about the motivation for participation in FOSS development are discussed. The results of the BCG study (Lakhani & Wolf 2005) and the FLOSS study (Ghosh 2005) will be presented in length. Selected results of different studies such as the Apache study of Roberts et al. (2006), the corporate wiki study of Majchrzak et al. (2006), the Linux kernel study of Hertel et al. (2003), the study of Hars an Ou (2002), the Freenet study of von Krogh et al. (2003), and the Canadian volunteerism study (Day & Devlin 1996) will be outlined in a shorter form. The FLOSS-US study (David, Waterman, & Arora 2003b), the FLOSS-Japan study (Mitsubishi Research Institute 2004d), and the FLOSS-Asia study (Mitsubishi Research Institute 2004c) are only briefly mentioned without referring to the results.

BCG study

In fall 2001 Lakhani and Wolf (2005) conducted a web-based study of FOSS projects at SourceForge (projects with status alpha, beta, and production/stable). They contacted 1530 FOSS developers and received 526 responses (Lakhani & Wolf 2005: 8). A second study in April 2002 contacted 573 developers of mature projects and received 158 valid answers.

The authors conclude that “enjoyment based intrinsic motivation” is the top single reason to contribute to FOSS projects (Lakhani & Wolf 2005: 12). This type of motivation is more important than extrinsic motivation such as better jobs and career advancement (Lakhani & Wolf 2005: 3). How creative a developer feels while working on his project is found to be the strongest and most pervasive driver for the number of working hours for a FOSS project (Lakhani & Wolf 2005: 16-17). The authors therefore conclude that “enjoyment-related intrinsic motivations in the form of a sense of creativity” has the most important impact on participation in FLOSS projects (Lakhani & Wolf 2005: 18).

The description of the items in the BCG study allows connecting the used categories with the categories used in this study:

- Enjoyment based intrinsic motivation is similar to fun seeking.
- The different items in the economic/extrinsic category of the BCG study can be assigned to investment in human capital, business requirement, personal requirement, and investment in reputation.

- The items in the category obligation/community can be assigned either to pro-social behavior or to peer recognition.

Motivation	Item	Reason to contribute[a]				Category comparision to this study
		All respondents	Volunteer contributors	Paid contributors	Difference[b]	
Enjoyment-based intrinsic m.	Coding is intellecturally stimulating	44.9%	46.1%	43.1%	n.s.	Fun seeking
Economic / extrinsic-based motivation	Improve programming skills	41.3%	45.8%	33.2%	3.56 / 0.00	Human capital
	Contribution needed oneself	58.7%	-	-	-	Business and personal requirements
	Contribution needed oneself (work related)	33.8%	19.3%	55.7%	10.53 / 0.00	Business requriement
	Contribution needed oneslef (non-work related)	29.7%	37.0%	18.9%	5.16 / 0.00	Personal requirement
	Enhance professional status	17.5%	13.9%	22.8%	3.01 / 0.00	Reptutation
Obligation / community-based intrinsic motivation	Believe that source code should be open	33.1%	34.8%	30.6%	n.s.	Pro-social
	Obligation to contribute becaue of use of FOSS	28.6%	29.6%	26.9%	n.s.	
	Dislike proprietary software and want to defeat it	20.3%	21.5%	18.5%	n.s.	
	Like working with development team	11.3%	11.5%	11.1%	n.s.	Peer recognition
	Enhance repuation in F/OSS community	11.0%	12.0%	9.5%	n.s.	

N=679

[a] Subjects could give up to three answers

[b] n.s. = not significant; t-statistic / p-value

Figure 43: Motivation in the BCG study[197]

If the categorization of this study is applied to the BCG study a ranking of different motivations to contribute to FOSS projects can be produced (see Figure 44). Fun seeking is the most important motivation to contribute followed by human capital, personal requirement, pro-social behavior, and finally reputation.

[197] Lakhani and Wolf (2005: 13-14).

Motivation in BCG survey[abc]	Importance ranking			Reason to contribute[d]		
	All	Volunteer	Paid[b]	All	Volunteer	Paid
Fun seeking	1	1	1	44.9%	46.1%	43.1%
Human capital	2	2	2	41.3%	45.8%	33.2%
Personal requirement	3	3	5	29.7%	37.0%	18.9%
Pro-social	4	4	3	27.3%	28.6%	25.3%
Reptutation	5	5	4	17.5%	13.9%	22.8%
Peer recognition	6	6	6	11.2%	11.8%	10.3%

[a] Based on the questioning the item could be clearly associated to the motivation categories in this survey.
[b] To make a comparision possible the category "work related contribution" is omitted. For volunteers this type of motivation is the sixth (out of seven) most important while for paid developers it is the most important).
[c] It should be noted that the methodology of the BCG and this study is different.
[d] For categories with more than one item the mean has been computed.

Figure 44: Raking of motivations in the BCG study[198]

FLOSS study

The FLOSS study involved 2774 subjects from which 2280 entries were entirely completed (Ghosh 2005: 30). The authors distinguished between social motivation, signaling, monetary concerns, political motivation, and product related motivation.

[198] Based on data of Lakhani and Wolf (2005: 13-14).

	Reason to particpate in FOSS development[a]		Motivation category FLOSS study	Motivation category in this study
	At the beginning	Today		
to learn and develop new skills	78.88%	70.47%	Social	Human capital
to share my knowledge and skills	49.84%	67.22%	Social	Pro-social
to participate in a new form of cooperation	34.54%	37.23%	Social	Peer recoginition
to improve OS/FS products of other developers	33.68%	39.83%	Signaling[d]	Pro-social
to participate in the OS/FS scene	30.56%	35.54%	Social	Peer recoginition
because I think that software should not be a proprietary product	30.15%	37.85%	Political	Pro-social
to solve a problem that could not be done by proprietary software	29.70%	29.61%	Product related	Personal requirement
to improve my job opportunities	23.87%	29.82%	Signaling[d]	Reputation[b]
to get help in realizing a good idea for a software product	23.83%	26.98%	Product related	Business requirement
to limit the power of large software companies	18.99%	28.87%	Political	Pro-social
to get a reputation in the OS/FS developers' scene	9.06%	12.03%	Signaling[d]	Peer recoginition
to distribute not marketable software	8.90%	10.01%	Signaling[d]	[c]
to make money	4.39%	12.31%	Monetary	Business requirement
I do not know	1.89%	1.52%	-	-
N	2438	2428	-	-

[a] Subjects could give maximally four answers.
[b] It may also be that job opportunies are increased by advancing human capital.
[c] It is not quite clear to which category of this study this question relates.
[d] The same category in the FLOSS study was also labeled "career".

Figure 45: Motivation in the FLOSS study[199]

The five above mentioned motivation categories were collapsed by the authors into four categories and the signaling category was renamed to career. The four different generic categories of motivation are:

- Social / community
- Career[200] / monetary concerns
- Political motivation
- Product related

[199] Ghosh (2005: 34) and Ghosh et al. (2002c).
[200] See comment d in Figure 45.

In an attempt to integrate different reasons to participate in FOSS development the authors of the study allocated each developer to one of their four motivation categories (Ghosh 2005: 34). Developers expressing *only* social motivation were allocated to the social motivation category (Ghosh 2005: 34). Developers who *additionally* expressed career / monetary reasons but *no* political or software related reasons were allocated to the career / monetary category. The ones who *additionally* expressed political motivations but *no* software related motivations were allocated to the former category. The remaining developers were allocated to the category software-related motivation. Because of this procedure roughly half of the subjects are motivated by social or community reasons, one third is motivated by career or monetary concerns, one-tenth is motivated by political reasons, and almost nobody is motivated by the product respectively software-related reasons.

Motivation category FLOSS study		**Motivation category in this study**			
Motivation	Percentage of all developers				
Social / community	53.2%	Human capital	Pro-social	Peer recoginition	-
Career / monetary concerns	31.4%	Pro-social	Peer recoginition	Reputation	Business requirement
Political motivation	12.7%	Pro-social	-	-	-
Product related	2.6%	Personal requirement	Business requirement	-	-

Figure 46: Comparison of motivation between the FLOSS study and this dissertation

The above mentioned approach does have some major problems:

Firstly, the following implicit assumptions are neither intuitively nor theoretically sound:

- There are no developers without social / community motivation.
- There are no politically motivated developers that are not to some extent motivated by career / monetary incentives.
- There are no developers motivated by the software itself that are not to some extent motivated by career / monetary incentives or that are not to some extent politically motivated.

Secondly, as can be seen in Figure 46 the motivation categories of the FLOSS study and the categories used in this study can hardly be compared. The category social / community is related to the categories human capital, pro-social behavior, and peer recognition. The category career / monetary concerns is related to pro-social behavior, peer recognition, reputation, and business requirement. Software related motivation is related to personal requirement and business requirement. The category political motivation is related to pro-social motivation. Unfortunately, the important motivation fun seeking seems not to exist in the FLOSS study.

An item included in the FLOSS study intends to measure selfishness respectively altruism (Ghosh 2005: 33-34). Developers were asked how they assess the balance between contribution to the FOSS community and the benefits from it. Roughly 9% of developers give more than they take which is considered to reflect altruism, 56% take more than they give which is considered to reflect selfishness, 15% give as much as they take, and 21% do not care or know (Ghosh et al. 2002c). The authors of the study conclude that these responses are consistent with the assumption that the motivation for FOSS participation is self-interested (Ghosh 2005: 33-34). This conclusion must be clearly challenged:

- *Firstly*, the answer options for the item on self-interest reflect the balance of contributions and benefits. This does not imply any information *why* a developer contributes. A politically motivated developer might benefit a lot from FOSS. Therefore, this developer would wrongly think to be self-interested. However, this developer would contribute even if he benefited nothing. As the authors of the study recognize themselves the item measures an outcome and not intent (Ghosh 2005: 33).
- *Secondly*, it could be hypothesized that term is "benefit" is interpreted very differently by selfish and altruistic developers (see also 5.5.2.3). The former may consider only reputation gains, human capital, and personal need as a benefit. In contrast to this interpretation altruistic developers may additionally consider the advancement of free software for the sake of mankind as a benefit. If so, altruistic person would score higher on the discussed selfishness scale.
- *Thirdly*, it could be assumed that selfish developers unintentionally overrate the worth of their contribution and underrate the worth of the benefits. The rational behind this argument is that the focus of a selfish person is on his own. The reversed argumentation may be applied to altruistic developers. Because their focus is on society they will perhaps underrate their own contribution. In consequence, altruistic developers would be categorized as selfish.
- *Fourthly,* The authors refined the study of selfishness by analyzing the relationship to the four motivational categories (Ghosh 2005: 33). Developers with a dominantly social/community motivation were the most selfish followed by developers with a political orientation. Only developers with product-related motivation are less selfish than the developers in the career/monetary category. This result seems to be very counterintuitive.
- *Fifthly, m*ore than half of all developers are considered to be motivated by social or community concerns. This contradicts the finding that most developers are self-interested. The motivational categories, the selfishness scale, or both are not reliable.

To sum up, while the FLOSS study itself is interesting the analysis of the data seems to be neither reliable nor comparable to the motivational typology used in this study.

Based on the FLOSS study several other studies were conducted which are only mentioned without presenting the results:

- The FLOSS-US study (David, Waterman, & Arora 2003b) received 1588 responses. It was conducted from January to June 2003 in English and Chinese with announcements in 9 different languages. Although the study's title suggest that only U.S. developers are surveyed this is not the case. More than half of the subjects came from Western Europe (53% David, Waterman, & Arora 2003b: 5). Only 27% of subjects came form North America.
- The FLOSS-Japan study (Mitsubishi Research Institute 2004d) was conducted between September and November 2003 in Japanese. The study is clearly dominated by Japanese developers that account for 99.8% of the 547 subjects in the sample.
- The FLOSS-Asia study (Mitsubishi Research Institute 2004c) was conducted between December 2003 and January 2004 in English, Traditional/Simplified Chinese, Korean, and Thai. It received 138 responses. Roughly 96% of developers in the study came from Asia. Korean, Thai, and Taiwanese

developers accounted for more than two-thirds. Together with subjects from China, Malaysia, Hong Kong, and India these developers accounted for more than 92% of the subjects in the sample.

Corporate wiki study

A study related to contribution to corporate wikis was conducted by Majchrzak et al. (2006).[201] Although such contributions are not directly connected to FOSS development the related problems are similar. The study seems to be especially interesting because it is accomplished in a corporate setting. Such quantitative studies can hardly be found for FOSS development.

The most important types of motivation to contribute to corporate wikis are personal requirements respectively work related requirements; the second most important motivation is peer recognition (Majchrzak, Wagner, & Yates 2006: 101). To improve reputation in the company or improve professional status were the least important motivations to contribute. However, the importance of reputation increases the more expert a contributor is.

The authors of the study suggest that there are four different types of contributors (Majchrzak, Wagner, & Yates 2006: 103):

- *Adders* add primarily new content
- *Synthesizers* integrate ideas, reorganize pages, and edit grammar or spelling
- *Multiplexers* add and synthesize content
- *Minimalists* refrain from contributing at all

Interestingly, the motivation to contribute is different for the different types of contributors (Majchrzak, Wagner, & Yates 2006: 103). This finding is very relevant for the main argumentation in this dissertation. *Synthesizers* are motivated by the frequency with which people access the wiki site and the impact on their reputation.[202] At the same time they are not motivated by personal requirements. In contrast, *adders* are motivated by easing their work, personal requirements and fulfilling their formal roles. However, they are not motivated by reputation gains.

Apache study

In a longitudinal study based on an interesting and sophisticated theoretical model Roberts et al. (2006) evaluate the interrelationships between motivation, participation, and performance of developers in Apache projects. Due to the focus on Apache projects generalizations should be made only carefully. It seems to be clear that the Apache projects are not representative for the vast amount of FOSS projects.

The authors found that contributors have multiple motivations to participate in FOSS projects (Roberts, Hann, & Slaughter 2006: 995-996). Some of these motivations are complementary whereas others are not. They

[201] For an empirical study on a similar topic see Peddibhotla and Subramani (2007). The authors researched the motivations of individuals to contribute to a public document repository. They found that self-expression, developing writing skills, enhancing the understanding of the topic, utilitarian motives, enjoyment, social affiliation, altruism, and reciprocity are important types of motivation.

[202] Unfortunately, the motivation of multiplexers and minimalists are not analyzed.

found no undermining effect of extrinsic motivation on intrinsic motivation (as an opposite example sees the Debian-Dunc-Tank story in chapter 3.4). At the same time, being paid for contribution is positively related to peer recognition and negatively related to personal requirements.

In the Apache study motivation has also an influence on the contribution level (Roberts, Hann, & Slaughter 2006: 996). Developers who are paid and developers motivated by peer recognition contribute more than developers motivated by personal requirement do. Intrinsic motivation does not have a significant impact on the contribution level.

Linux kernel study

In a study on the motivation of FOSS developers Hertel et al. (2003) surveyed 141 contributors to the Linux kernel. To understand motivational processes the authors contrasted in a very inspiring approach two different theoretical models originating from social science.

The *first model* is based on research on voluntary action in social movements. Based on Klandermans (1997) the authors of the study recognize the following motives (Hertel, Niedner, & Herrmann 2003: 1163):

- ***Collective motives*** are based on the evaluation of the movement goals weighted by the perceived likelihood that these goals are reached. This category is similar to pro-social motivation in this study.
- ***Social motives***[203] are based on the expected reactions of significant others such as friends and family member. The motivation is the higher the more positive the expected reactions of significant others are. These reactions are weighted by the perceived importance of the corresponding significant others.
- ***Reward motives*** result from expected costs and benefits. Costs may be time and money and benefits may be personal requirements, reputation effects, fun, or the enhancement of human capital. These costs and benefits are weighted by their likelihood.
- ***Identification motives*** are based on the involvement in a sub-group itself. This motivation is similar to peer recognition in this study.

The authors found that the motivational processes within the Linux kernel development have similarities to the ones in social movements (Hertel, Niedner, & Herrmann 2003: 1174-1175). Collective motives, social motives, reward motives (fun, personal requirements), and identification motives were altogether important motivational factors that correlated positively with invested time in Linux related activities.

The *second model* is based on motivational processes found in small team collaboration. In this model the motivation of individuals to work in a virtual team is determined by four components (Hertel, Niedner, & Herrmann 2003: 1164-1165):

- ***Valence*** defined as the subjective evaluation of team goals
- ***Instrumentality*** defined as the perceived importance of indispensability of one's own contributions for the group outcome
- ***Self-efficacy*** defined as the perceived capability of showing the required activities for the team task

[203] Because Hertel et al. (2003: 1163) consider social motives as misleading they call this motivation norm-oriented behavior. However, this seems to be even more misleading.

- ***Trust*** defined as the expectancy that team members will reciprocate one's contribution, that one will not be exploited by other team members, and that the electronic support system works reliably

Again, the results show that the motivational processes found in small teams in business organizations can partly be found as well in Linux development. The activities of participants in Linux kernel development were determined by valence, instrumentality, and self-efficacy. Contrary to the theoretical model trust played only a minor role concerning the motivation in Linux development teams.

A very interesting aspect in the Linux study is that motivational goals are weighted. Collective motives are weighted by the perceived likelihood that they are reached, for social motives the reactions of others is weighted by their perceived importance, and for reward motives the costs and benefits are weighted by their likelihood.[204]

It is argued here that such weighing is less important for fun seeking, paid development, and investment in human capital but more important for pro-social motivation, personal requirement, and investment in reputation:

- Whether one has fun i) is very fast detectable and does ii) hardly depend on other developers. It seems therefore that the likelihood that a fun seeker spends a long time for a task that is not fun is low. Similarly, that nothing is learned by contributing to a FOSS project seems to be unlikely. That the salary of a paid developer is not paid out and that an employer gets insolvent may happen but is unlikely.
- In contrast, whether a contribution to a FOSS project enhances the freedom of software, whether a contribution meets a personal requirement, and whether a valuable signal can be produced is more questionable. Producing a valuable signal i) takes a long time and whether doing so succeeds depends ii) to some extent on other developers competing for the same signal. A developer who starts contributing to the Linux kernel may hardly be sure that after several years of contribution he will get a reputation as a good hacker. Similarly, that a personal requirement is satisfied is not sure. It may turn out that a desired functionality is technologically too difficult to implement. A developer driven by pro-social motivation can also not be sure whether the project he contributes to will be after all a cornerstone of FOSS.

Hars and Ou study

In their study Hars and Ou (2002) surveyed the motivation of FOSS developers. They distinguished between internal and external factors. They consider intrinsic motivation, altruism, and community identification as internal factors. External rewards are split in future rewards and personal needs. Future rewards are revenues from related products and services, human capital, self-marketing (investment in reputation), and peer recognition.

The most important motivation of developers is investment in human capital followed by intrinsic motivation and peer recognition (see Figure 47). Interestingly, for salaried and contract programmers intrinsic motivation is more important than investment in human capital while for students and hobby programmers human capital is more important than intrinsic motivation. Programmers paid for their contributions to FOSS projects differ

[204] This weighing is closely related to self-efficacy and instrumentality.

slightly from the two aforementioned subgroups. For such developers self-marketing (investment in reputation) as well as selling products and services are more important than peer recognition.

Percentage of subjects who rank very high or high on a specific motivation										
	Motivation	**All**		**Students and hobby programmers**		**Salaried and contract programmers**		**Programmers paid for FOSS development**	
		Percent	Rank	Percent	Rank	Percent	Rank	Percent	Rank
Internal	Intrinsic motivation	**79.7%**	**2**	**81.8%**	**2**	**92.6%**	**1**	**61.5%**	**3**
	Altruism	16.5%	7	24.2%	7	11.1%	7	7.7%	8
	Community identification	27.8%	6	36.4%	4	18.5%	6	30.8%	7
External	Selling products	13.9%	8	6.1%	8	3.7%	8	53.8%	4
	Human capital	**88.3%**	**1**	**96.6%**	**1**	**88.5%**	**2**	**84.6%**	**1**
	Self marketing	36.7%	5	33.3%	6	29.6%	5	**69.2%**	**2**
	Peer recognition	**43.0%**	**3**	**42.4%**	**3**	**48.1%**	**3**	46.2%	5
	Personal need	38.5%	4	36.4%	4	38.5%	4	38.5%	6

Figure 47: Motivation in the Hars and Ou study[205]

Freenet study

In a study of the Freenet project von Krogh et al. (2003) research how developers make their initial project contribution.[206] Von Krogh et al. (2003: 1220) classified the participants of the Freenet developer list as *non-developers*, *joiners* who do not have access to the CVS repository, *newcomers* who have just begun to make changes to the CVS repository, and *developers* who also contribute code to the project but have moved beyond the newcomer stage.

Based on the collected data von Krogh et al. (2003: 1229-1234) propose that that joiners who get newcomers follow a joining script (see also von Krogh & von Hippel 2003: 1154-1155). Contributors who follow this joining script are more likely to be granted access to the developer community than those who do not follow it. Part of the joining script is for example reporting specific technical bugs instead of giving general feedback in the mailing list. According to the joining script, after being admitted into the community the newcomers specialize in mundane tasks before they start contributing to work that is more complex. Newcomers often give early gifts in terms of software modules and features. This is rewarded and newcomers can contribute to the project by continuing to work on those modules or features that they know best.

[205] Hars and Ou (2002: 34).

[206] Interestingly, the authors adopt in difference to most studies on FOSS development a qualitative grounded approach to develop analytical categories and propositions.

Canadian volunteerism study

In a large scale Canadian study it was found that demographic profiles of individuals are related to volunteerism (Day & Devlin 1996: 44-47):

- ***Gender***: Males are less likely to volunteer than females but if they do they spend more hours.
- ***Marital status***: Married individuals do more likely become volunteers.
- ***Age***: The relationship between age and the probability to volunteer is nonlinear. The probability increases throughout the middle ages but decreases after the age of 65.
- ***Education***: The propensity to volunteer increases with education, especially for women.
- ***Religious beliefs***: The strength of religious beliefs is positively related to volunteering.
- ***Health***: The state of individual's health influences the propensity to volunteer.
- ***Household income***: If annual household income rises beyond 20'000$ the probability of becoming a volunteer unambiguously rises.
- ***Employment***: The employment status is an important determinant of the decision to volunteer. Full-time workers are less likely to donate their time than those working part-time or working not at all.

Summing up empirical evidence

Summing up it can be concluded that having fun and investment in human capital are the two most important factors to contribute to FOSS development. Peer recognition, pro-social motivation, and personal requirements are clearly less important. Contrary to scientific writings investment in reputation does not seem to play any significant role in motivating developers to contribute.[207]

3.4 Difficulties of working together

Torvalds and Diamond (2001: 163-164) argue that the involvement of commercial firms in Linux development is beneficial because they accomplish some of the less interesting and boring work such as maintenance and support. The same argumentation holds true for paid developers. They accomplished work that would not be done otherwise. Additionally, because of these commercial firms Linux may flow into new markets. Therefore, one may argue that commercial firms and paid developers are beneficial for the advancement of a FOSS project.

However, there are difficulties between the community and commercial firms respectively between non-paid and paid developers.

[207] However, it must be noted that investment in reputation is perhaps i) the socially least desirable answer option and ii) more abstract than other motivation types. This may influence the measured importance of investment in reputation.

3.4.1 Community vs. commercial firms

The very basic problem between commercial firms and the FOSS community is that in most cases the reasons for contribution are different. Commercial firms are primarily interested in making profits while the community wants to foster free software or to write a sound application.

In a typology describing the relationship of commercial firms to FOSS communities Dahlander and Magnusson (2005: 487) distinguish between *symbiotic*, *communalistic,* and *parasitic* firms:

- ***Parasitic*** firms focus only on their own benefit without concerns whether actions harm the community or not. They do not obey community values and rules deliberately. For the management the only task is to avoid direct conflicts with the community.
- In the ***communalistic*** approach firms try to benefit from the community without harming it. In this approach there is no direct involvement in the community itself. The firm devotes manpower to a FOSS project to get problems fixed that are essential for the firm but would not be fixed otherwise.
- In the ***symbiotic*** approach firms consider the effects of their behavior on the FOSS community (Dahlander & Magnusson 2005: 488). Norms and rules of the community are respected. This is demanding and consumes scare resources. The community itself is part of the knowledge base of the firm similar to communities of practice (Wenger & Snyder 2000). Both in FOSS projects as in communities of practice there is the problem of diametrically opposed needs for openness on the one and control on the other hand (Dahlander & Magnusson 2005: 488). To handle this problem the management needs to be directly and actively involved in community development.

Firm approaches toward the FOSS community			
	Symbiotic	**Commensalistic**	**Parasitic**
	Community gains because of the firm involvement	Community neither gains nor loses because of the firm involvement	Community loses because of the firm involvement
Nature of relationship	Giving something to the community, often through a firm-established community	Search for useful input from the community	Search for useful input without obeying norms, values and rules
Possibility of influencing community	High	Low	None
Managerial challanges	Respect norms and values Obey licenses Resource consumption of developing community	Respect norms and values Obey licenses Getting acceptance of the community for using ist resources in commercial applications	Avoiding direct conflicts
Operational means of subtle control	Attracting developers Aligning different interests Resolving ambiguity about control and ownership Devoting personnel to work in communities Creating and maintaining reputation Fringe benefits Interaction tools Selling development tasks	Devoting personnel to work in communities	None

Figure 48: Firms approaches towards the FOSS community[208]

Commercial firms that intend to be estimated by the FOSS community have to consider two points. *Firstly*, they have to understand the community and FOSS development (Feller & Fitzgerald 2002: 6). Michael Tiemann, at that time CTO of RedHat, stated in an interview (Dougherty, Sims, & Tiemann 2000):

> *People who think money is the engine are wrong. Money is just the gas, and the engine is the open source development community. And the quality of the people who are in that community determines how much horse power this movement's going to have.*

---

[208] Dahlander and Magnusson (2005: 487).

Secondly, they most conform to the values of the FOSS community (Bonaccorsi & Rossi 2004). Additionally, they must cooperate by *gifting* their code and they must not hijack the source code. That *at least some* commercial firms care about their reputation can be recognized by the result of Henkel's (2006) study. For commercial firms the most important reason to reveal source code besides legal concerns is the intent to be a good player in the FOSS community (Henkel 2006: 961). This shows that some commercial firms are interested in working together with the FOSS community.

The values of sharing and openness found in FOSS development are not necessarily but at least sometimes in conflict to the goal of firms to generate profits. Certainly, to find a balance between being a good player in the community and the firm's ambition to generate profit is not always easy (Dahlander & Magnusson 2005: 484). If a firm tries to exert too much control over development it is questionable whether the FOSS project will generate the energy, interest, and creativity that are needed. FOSS projects member are free to leave or fork the project whenever they want. They will do so if a commercial firm does not play according to the rules. However, with too little control and direction it may be that the project may go in another direction than desired by the firm.

Problems of working together: The NetBSD example

An example of a troublesome relationship between the FOSS community and a commercial firm is the NetBSD project (for another example see the Mambo-Joomla! case in chapter 2.7). According to a co-founder of the NetBSD project one reason for the decline of the project was the heavy influence of the commercial firm Wasabi (Biancuzzi 2006; Hannum 2006). Development and release management was adjusted to meet Wasabi's requirement that was not always beneficial for the NetBSD project. This was possible because people in charge in the project were also employees of Wasabi. At the same time Wasabi was not contributing source code to the project but hired NetBSD developers that were lost to the project. Hannum concluded that Wasabi can do whatever they want in the NetBSD project and there is no check and balance. Consequently, Hannum states that the project "has stagnated to the point of irrelevance".

3.4.2 Paid vs. non-paid developers

There was some fear in the Linux community when the founder of Linux Linus Torvalds moved from the University of Helsinki to the commercial firm Transmeta (Moody 1997). There were worries that Torvalds' new job does change his relationship to the community. This was not the case. In fact, it was the birth of his first daughter that had an influence on his engagement in Linux development.

Based on the example of Torvalds one may argue that Lakhani et al. (2005: 19) correctly conclude that neither extrinsic nor intrinsic motivation destroys or dominates the efficacy of the other. They observed in their study rather that there is an interplay between extrinsic and intrinsic motivation. In fact, Franck and Jungwirth (2003b) argued that FOSS managed to have succeeded in creating a governance structure that attracts rent-seekers (e.g. reputation investors) as well as donators (pro-social motivation). They argue that "bringing people with different motivations, rent-seekers and donators, into a symbiotic relationship is the key innovation in the governance structure of open source" (Franck & Jungwirth 2003b: 419).

However, as we know fairness matters (Cropanzano et al. 2001: 183; Rabin 1993; Van Prooijen, van den Bos, & Wilke 2004: 33). Paying certain developers and others not may be a sensitive issue. If developers consider that fairness is violated they may react by stop contributing to the project. There are two lines of argumentation why fairness norms may be violated by paying FOSS developers (see also chapter 5.4.2.1.2). *Firstly*, procedural justice may be violated (Thibaut & Walker 1975). Most probably there are more good developers deserving payment than available money. In consequence, a selection of developers that are paid must be made. The corresponding fair selection procedures are not easy to establish. In contrast, paying nobody is certainly easier to implement. *Secondly*, distributive justice may be violated (Adams 1963). The equality rule states that everyone should be treated equally respectively that everyone should get exactly the same share (Deutsch 1975).[209] By paying some developers but not all this rule is clearly violated.

In chapter 4.3.7 it will be argued that leaders in FOSS projects are accepted to a big extent because of their *expert power*. To get such power on has to write beautiful source code. Leaders are accepted as long as their source code is admired. The strong meritocracy in FOSS projects is based on this characteristic. By giving project leaders the possibility to pay some developers they get automatically *reward power*. Paid developers get dependent on the leaders goodwill. In consequence, meritocracy may be threatened. One may think that a leader deprive critical developers of their salary.

The problems of payment are probably much higher if the FOSS project itself pays developers. If commercial firms pay some developers i) the questions of fair procedures and fair outcome are settled outside of the project and therefore not subject of a grievance within the project itself and ii) the power to reward are not hold by project leaders.

Problems of working together: The Debian-Dunc-Tank example

In the following, the Debian-Dunc-Tank example is presented to show the problems that may emerge between non-paid and paid developers. Debian is a widely used and known Linux distribution. It is the basis of several known other Linux distributions such as Knoppix or Ubuntu. While Debian is certainly tremendously important for the FOSS community it has also a reputation for very long consultations and decision making processes, long release cycles, as well as a somewhat chaotic development process. Although it is a community project commercial firms such as HP have an interest in the successful development (Shankland 2006). The projects members are to a very great part volunteers (Jaspert 2006). Commercial firms from the outside pay some developers but the Debian project itself does not pay anybody.

In an experiment called Dunc-Tank[210] a group of individuals heavily associated with the Debian project itself try to accelerate the newest release 4.0 called the “etch”. They do so by funding two release managers respectively paying them full-time. In fact, the funding of two developers drew some attention (Slashdot 2006) and angered unpaid developers (Thurston 2006). A group of developers pointed to the problems of the experiment and asked in an open email for what the release managers are paid for, why the selected individuals are paid and other’s doing similarly important work not, and how the output of the two developers is controlled (Jaspert 2006).[211] They are mainly concerned that the experiment establishes a two-class project

[209] Additionally, the equity rule that individuals should be paid according to their input may also be violated. As argued there are most probably more hackers who contributed valuable source code than money.

[210] Dunc is an acronym standing for "**D**evelopment **u**nder **n**umismatic **c**ontrol" (Dunc-Tank 2006). As stated by the project it could be equally called “coin-operated coding”. Dunc also refers to an occurrence at an auction of the Electronic Frontiers Australia in 2003.

[211] See Towns (2006) for a reply.

in which some are paid and others not. A comment on Slashdot states similarly that "dunk-tanc.org really is splitting the community" (Slashdot 2006). Another individual comments that there is a preference "…that the people contributing to Debian are motivated by the desire to solve problems, and to make a good product better; as opposed to having Debian be contributed by programmers whose attitude is 'whatever, fuck it, it's good enough; where's my ten bucks?'". The group of developers mentioned above fear that the experiment destroys the motivation of the volunteers not paid (Jaspert 2006). They argue that the problem is not that developers are paid for their work (which is done in the Debian project for a long time) but that this is done by an institution tightly connected to the project itself. They further argue that the experiment "… demotivated a lot of people who now either resigned, simply stopped doing (parts of their) Debian work or are doing a lot less than they did before …". Thurston (2006) similarly states that because of the Dunc-Tank experiment the release processes in fact slowed down. Whether the Dunc-Tank really influenced the project negatively cannot yet be finally judged. However, the example may be a showcase how sensitive the FOSS community is to unfair processes or dangers to meritocracy.

3.5 Summing up

The basic insights of this chapter are:

- The motivational structure of an individual (chapter 3), the circumstances and the structure of the decision problem (chapter 4 and chapter 5), and the decisions of other individuals (chapter 6) may lead to special dynamics (chapter 7) and are related to each other. > See chapter 3
- In self-determination theory there are two basic types of motivation: intrinsic and extrinsic motivation. The former refers to an activity that is inherently interesting while the later refers to a desirable outcome. > See chapter 3.1
- Individuals driven by intrinsic motivation are called in this study fun seekers. Intrinsic motivation is in particular import for creativity. There are three human needs that foster intrinsic motivation: autonomy, competence, and relatedness. In FOSS development autonomy is especially high. Developers are completely free to choose the tasks they want to accomplish. > See chapter 3.2.1
- There are different types of extrinsic motivation related to FOSS development. It is claimed that the following types are relevant: *Pro-social motivation* is based on norms towards keeping software free. *Peer recognition* is related on reputation within the FOSS community. *Paid developers* engage in FOSS development because of the monetary reward. Developers with a *personal requirement* contribute to have a functionality implemented or a bug fixed. *Investors in reputation* want to signal their capabilities to get a better job. *Investors in human capital* want to enhance their coding expertise to get a better job. > See chapter 3.2.2
- The Linux kernel study suggests that motivational goals should be weighted by the perceived likelihood that these goals are reached. It should for example not be assumed that investors in reputation will manage to produce a valuable signal. It was suggested that such a weighing is especially important for personal requirement, investment in reputation, and pro-social motivation. > See chapter 3.3
- Empirical evidence about the motivation of developers suggests that having fun and investment in human capital are the two most important factors to contribute to FOSS development. Peer

recognition, pro-social motivation, and personal requirements are clearly less important. Investment in reputation does not play a significant role in motivating developers. > See chapter 3.3

- The relationship between paid and non-paid contributors respectively commercial firms and the community may be troublesome. > See chapter 3.4

Chapter 3 introduces motivation as a basic part of the argumentation. It is thought that motivation is very relevant for solving social dilemmas (see chapter 5) but also for the advancement of FOSS projects (see chapter 8). There are two crucial lessons for the further advancement of this dissertation. *Firstly*, the motivation of FOSS developers is very multifaceted. Different developers are differently motivated *Secondly*, contingent on the type of motivation individuals respond in an other way to certain governance structures and mechanisms. While for example autonomy is theoretically important to keep fun seekers contributing it is not important for individuals motivated by personal requirement.

4 Governance structures and mechanisms: The promoting or restricting forces

Smith (1976: book I, chapter 1, p. 7-16) suggested that the division of labor increases productivity.[212] Similarly, FOSS development takes heavily into account that distributing work can boost productivity exponentially (Goetz 2003). Therefore, FOSS projects need adjusted governance structures and mechanisms to create it (Weber 2004: 186). These structures and mechanisms are special. Mann (2006: 1-2) states that FOSS solutions themselves are "not as provocative as the way in which they are developed and distributed". In fact, the structures and mechanisms do not seem to be appropriate at first sight. Von Krogh and von Hippel (2006: 975) state concerning FOSS development: "Who would ever have imagined that innovation could flourish under conditions like those?"

The consequence of these special structures and mechanisms is that someone who wants to use the FOSS development model needs to develop "a nontraditional model of funding and project management" (PITAC 2003: 7). At the same time these nontraditional structures and mechanisms of the FOSS development model seem to be one of the key elements of its success. About 32% of developers in the FOSS-US study strongly agree with the statement that a reason to start developing FOSS is that it is the best way to develop software (David, Waterman, & Arora 2003b: 19).

Two main characteristics of FOSS development are in sharp contrast to most other production processes:

- *Firstly*, communication and coordination processes are very special. For communication and coordination in FOSS projects emails and other electronic communication tools are used but almost no traditional methods such as meetings, phone calls, stable plans, system-level design, schedules, and defined processes. FOSS projects are extreme cases of geographically distributed development in which developers rarely or never meet each other face-to-face (Mockus, Fielding, & Herbsleb 2002: 310). Herbsleb et al. (1999: 85) argue that coordination mechanisms get especially important in such geographically distributed development.
- *Secondly*, most FOSS project members are not paid for their contributions and therefore not subject to hierarchical subordination. Governance structures and motivations must then be adjusted to this circumstance respectively to the special types of motivation. It is mostly assumed that FOSS developers follow a leaders' advice based on the belief that they have exceptional coding or other

[212] In the known example of pin-making a sole individual produces below 20 pins a day or "perhaps not one pin in a day" (Smith 1976, book I, chapter 1, 8-9). If the production process is divided into about 18 distinct operations each of ten individuals performing only one, two, or three distinct operations can produce about 4800 pins a day. Since the time of Smith around 1776 this productivity increased steadily. Karl Marx (1919: 425, part 424, chapter 413, section 428) estimated that around 1867 an individual might produce almost 600'000 pins a day (An individual can supervise 4 pin making machines that produce 145'000 pins a day). Based on data provided by Pratten (1980: 95) one can estimate that back in 1980 the output of a single employer in a U.K pin making company was about 5.76 Millions pins a day (An individual can supervise 24 pin making machine that produce 500 pins a minute; it is assumed that an employer works 8 hours a day).

qualities. In this sense, FOSS leaders must have a kind of charismatic authority as described by Max Weber (1990: 654-681; see also Yukl 1998: 298-323).

A problem of researching governance structures and mechanisms of FOSS projects is that they are mostly neither formal (Weber 2004: 185) nor explicitly recorded. Studying a phenomenon that is informal and hardly explicitly described is methodologically very demanding.

Butler (2001: 347) argues that neither providing a room for reception on an academic conference does assure social interaction nor does providing electronic infrastructures guarantee the emergence of social activity. Starting a FOSS project and adopting all of the structures and mechanisms discussed in this chapter does not necessarily lead to success. Certainly, proper structures and mechanisms help but furthermore individuals must want to cooperate.

Organizational structure may be the result of FOSS developer's motivation. At the project start members are able to configure the project structure at their convenience. However, organizational structures also influence motivation. Organizational characteristics such as credit files facilitate for example the proper functioning of signaling mechanisms while a functioning peer review fosters learning. Ghosh (2005: 27) argues that the motivation getting to the top of a hierarchal organization is different from doing so in a flat organization. Following the wording of Chandler (1962) as well as of Hall and Saias (Burgelman 1983) developers' motivation follows FOSS project structure *and* FOSS project structure follows developers' motivation.[213] In consequence, the breakdown of FOSS project structures and mechanism is important to understand the motivation to contribute to FOSS projects and vice versa (see also the corresponding argumentation in chapter 3).

In the following subchapter governance structures and mechanisms are discussed in general. To disclose the relevant problems of governance structures and mechanisms in FOSS a good and a bad example are presented. Afterwards, the different structures and mechanisms are discussed in detail. The subsequent subchapter identifies examples beyond software development in which FOSS governance principles can be found as well. The reason for doing so is to show that the phenomenon analyzed in this dissertation is important beyond software development. Because this is thought to be very important it is put to extreme in the subchapter about the self-governance of pirate communities. Although pirating has in content nothing to do with FOSS development the governance structures and mechanisms such as fairness, meritocracy, and self-governance are quite similar.

4.1 Governance structures and mechanisms: A theoretical overview

In Williamson's (1975; 1991) transaction cost theory there are three different generic forms of economic organization:

- In ***markets*** transactions are based on discrete contracts (Ring & Vandeven 1992: 485). These transactions are relatively short-term transfers of property rights between autonomous buyers and sellers. In markets there are sharp-in and sharp-out transactions (sellers dept of performance and

[213] Chandler (1962) claims that the structure of an organization follows its strategy. Others such as for example Hall and Saias (1980) argued that strategy follows structure. Depending on which part of the strategy process is observed both "strategy follows structure" and "structure follows strategy" are correct (Burgelman 1983: 67). In consequence, one may argue that "structure follows strategy follows structure".

buyer's dept of payment are unambiguous). The exchanged property, products, or services tend to be non-specific and transacted among many traders. The competitive market and classical contract law provide an efficient safeguard to the involved parties.

- In ***hierarchal settings*** transactions deal with the production of wealth or rationing of resources between superiors and subordinates (Ring & Vandeven 1992: 485). Hierarchical transactions rest on the legal right of the owner or employer to issue commands and on the duty of obedience by the employee.[214] Conflicts are in consequence resolved by authority that is rooted in the asymmetrical distribution of command and obedience roles. In such a transaction setting it is necessary that there is a social agreement that hierarchy has the legitimate authority (Ouchi 1980: 130). Hierarchy is thought to be the proper governance structure if outcomes are uncertain, if transactions recur frequently, and if transactions require unique or specific investments.
- ***Hybrid forms*** of organization are situated between markets and hierarchies and include franchising, joint ventures (Garrette & Quelin 1994; Williamson 1985: 83), clusters, networks, symbiotic arrangements, supply chain systems, administered channels, nonstandard contracts, collective trademarks, partnerships, cooperatives, alliances, virtual firms, and clusters of venture capitalists (Menard 2004: 347-350). As Garrette and Quelin (1994: 395) argue hybrid forms are conceptually not defined per se but only by difference to markets and hierarchies.

There has been some critic that transaction cost theory mainly deals with the polar forms and that it neglects the hybrid form (Ouchi 1980: 131; Williamson 1991: 269). A reason for this observation is probably that markets and hierarchies are theoretically well established while there are conceptual problems of defining hybrid forms.

Another critique on Williamson's transaction cost theory is his central assumption of opportunistic behavior (Ghoshal & Moran 1996: 14). Opportunism includes behavior such as lie, cheat, mislead, disguise, obfuscate, feign, distort, and confuse (Williamson 1985: 51). Based on this assumption it seems to be questionable how FOSS projects could exist. There are always golden opportunities in which individuals can be sure to get away with a norm violation. Additionally, Ghoshal and Moran (1996: 14) argue that this assumption may result in a self-fulfilling prophecy. It seems to be clear that this assumption about opportunistic behavior is sometimes but clearly not always suitable.

Drucker (2001) argues that knowledge will get more and more a key resource and that the knowledge workers will be dominant in a company. If this assumption is true people should be treated as an asset to be developed and not as a cost factor that has to be controlled (Bartlett & Ghoshal 1996: 345). In knowledge intensive companies the most valuable knowledge is not anymore held by the top managers but by teams of professional staff (Child & Rodrigues 2003: 344). Software companies are certainly knowledge intensive. Software is about people and software development is a social process (Messerschmitt & Szyperski 2003: 4-5). It is the expression of wishes and action of human programmers. Successful software is on the on hand the result of technical proficiency but on the other hand the result of a deep managerial, social, and psychological

[214] For example, the Swiss Code of Obligations demands in Article 321d(2) that "the employer may establish general directives and give special instructions about the executive of work and the conduct of employees in the enterprise or in the household. The employee must in good faith observe such general directives and specific instructions so given to him" (Schweizerische Eidgenossenschaft 2006; translation from: Swiss-American Chamber of Commerce 1995: 95).

understanding of what users need and want. For such knowledge intensive tasks the transaction cost theory approach seems to be of minor usefulness:

- ***Hierarchies*** request that superiors know how a task should be accomplished. In software development this is hardly the case. Superiors can hardly judge the work of subordinates.
- Software is a very complex product with many uncertainties. Therefore, ***markets*** seem not to be suitable for software production.

The structures and mechanisms found in FOSS projects are additionally neither comparable to the ones in hierarchies nor to the ones in markets. Few software developers in FOSS projects are under control of hierarchical subordination. Market relations that involve sellers and buyers are also not characteristically for FOSS projects.

It can be concluded that the theoretical frame of the transaction cost theory is not completely suitable for describing governance structures and mechanisms of FOSS development.

If not transaction cost theory: What else?

Because of the critique on Williamsons' transaction cost theory in general and especially about hybrid organizations there have been different attempts to enlarge the reflections on governance structures and mechanisms:

- Ring and van de Veen (1992: 487) argue that there are next to markets and hierarchies two further types of contracting, namely ***recurrent*** and ***relational contracts***. For both types of contracts trust, norms, and reciprocity are important elements (Ring & Vandeven 1992: 491-492). An example of a related transaction mechanism is to te-uchi (clapping of hands) in Japan to govern relationships (Ring & Vandeven 1992: 495).
- Ouchi (1980: 132) observes that Japanese firms rely to a great deal upon inexperienced workers and socialize them until they accept the company's goal as their own. It is then not necessary anymore to control performance because the employee's inclination is to do what is best for the firm.[215] Important is the community based acceptance of the legitimacy of the institution. Based on Durkheim's (1902) reflections on organic associations Ouchi (1980: 132) argues that socialization is a principal mechanism of control.[216] He calls a governance structure based on this principle a ***clan***. It is thought that organizations in the clan form are typically technology advanced or closely integrated industries where teamwork is common, technologies change often, and individual performance is highly ambiguous (Ouchi 1980: 136).
- Osterloh and Frey (2000) argue that there is next to markets and hierarchies another coordination mechanism that is based on participation and ***self-organization***. Self-organization is also called a grown order, a self-generating or an endogenous order, or a spontaneous order (Hayek 1978:

[215] For an overview over performance controlling of people in organizations see Kerr (1981). For an overview over designs for organizational control see Dunbar (1981).

[216] Drawing on sociological classics one could also refer to Tönnies' (1912) concept of Gemeinschaft (community).

37).[217/218] Such a spontaneous order may be so complex that no one can comprehend it in its totality (Hayek 1978: 41). Hayek (1978: 46) argues that in the case of spontaneous order nobody does have to know every aspect that in turn makes it especially suitable for complex circumstances. Similarly, in biology self-organization is thought to have the advantage that it does not overextend leaders and that it fosters flexibility (Camazine 2001: 63-67).

- For Ostrom (1997: 8-18) there are theoretically three possible governance structures to solve social dilemmas: i) a ***Leviathan*** respectively a hierarchical governance structure, ii) ***privatization and market mechanisms***, and iii) ***self-governance*** of communities. It is often implicitly assumed that an important aspect of self-governance is democratic decision making (Townsend 1995: 40). For Ostrom (1997: 93) a collective choice arrangement is an important design principle of long enduring common pool resources. Most individuals affected by operational rules should be able to participate in its modification. Proper control and monitoring is another important element of self-governing communities (Ostrom 1997: 94). However, it is mutual monitoring and peer control and not hierarchical monitoring (Child & Rodrigues 2003: 352).

As proposed by various authors there seems to be governance structures and mechanisms that are different from either markets or hierarchies. The presented alternatives are based on social norms, embeddedness in social networks, and self-governance. Opportunism is *a* but not *the* dominant behavior. In the following governance structures and mechanisms as described in this subchapter are considered to be relevant for FOSS projects.

4.2 Linux vs. NetBSD: A manifest comparison

In the following Linux as an example for successful and NetBSD as an unsuccessful example for governance structures and mechanisms are presented.

4.2.1 Linux

Bollinger et al. (1999: 89) states that Linus Torvalds is not the only person ever who initiated a large and complex FOSS project. Nevertheless, Linux could generate higher levels of growth and application diversity than most other FOSS projects. Bollinger et al. (1999: 89) ask in consequence “Why is that”? Without being able to answer the question completely it is attempted to shed some light on it.

Raymond (1999a: 37-40) writes about Linux and Linus Torvalds:[219]

---

[217] Self-organization refers to a special order and structures of physical, biological, and social systems based on pattern formation through internal interaction without intervention by external directing influences (adapted from Camazine 2001: 7).

[218] In contrast to this arguing Weber (2004: 132) states that the self-organization in FOSS projects is a myth. He argues *firstly* that self-organization is used too often as a placeholder for an unspecified mechanism. *Secondly*, for Weber the term self-organization is normatively connoted. These arguments have to be taken seriously.

[219] According to Bollinger et al. (1999: 89) another factor for the success of Linus Torvalds was mere luck. Torvalds was lucky to start development at the time the Internet revolution took off and that he was able to build upon a hacker tradition of twenty years. Additionally, the refusal of Tanenbaum to accept patches for Minix was probably also very beneficial for Linux.

> *In fact, I think Linus's cleverest and most consequential hack was not the construction of the Linux kernel itself, but rather his invention of the Linux development model. … Granted, Linus is a damn fine hacker. … But Linux didn't represent any awesome conceptual leap forward. Linus is not (or at least, not yet) an innovative genius of design in the way, say, Richard Stallman or James Gosling (of NeWS and JAVA™) are … . Linus was keeping his hacker/users constantly stimulated and rewarded—stimulated by the prospect of having an ego-satisfying piece of the action, rewarded by the sight of constant (even daily) improvement in their work.*

Bezroukov (2004) observes similarly that Linux was for a long time technically weaker than the operating systems of the BSD family such as FreeBSD, OpenBSD, or NetBSD. He argues that "technically speaking Linux is a pretty boring, conservative reimplementation of Unix" and "that during the first ten years of development "definitely failed to surpass in quality FreeBSD". According to Bezroukov the success of Linux is based on the social achievements and on its collaborative Internet-based development. He claims that the development model of Linux differed from Minix or the BSD family in the following: [220]

- The ***license*** of Linux is the GNU GPL. The big difference to the BSD license used for the BSD operating systems (e.g. FreeBSD, OpenBSD, or NetBSD) and the proprietary license of Minix[221] is its copyleft mechanism. As already discussed (see 2.5.1.2) copyleft means that "anyone who redistributes the software, with or without changes, must pass along the freedom to further copy and change it" and it requires that "all modified and extended versions of the program to be free software as well" (FSF 2004). The importance of the GNU GPL license and the copyleft mechanism for Linux may be attributed to different rationales: *Firstly*, licenses are a sort of constitution (Weber 2004: 179). The BSD license has few constraints and is not very detailed (Weber 2004: 181). The GNU GPL is more explicit about political and social principles. The main message is to keep developers away from hoarding software code (Weber 2004: 182). You can take the software but you must give everything back.[222] Such a constitution fosters conditional cooperation and reciprocity in contrast to the BSD license. *Secondly*, the commercial motivation to fork a project is lower under the GNU GPL than under the BSD license. In consequence the danger of having different incompatible software solutions such as in the in the commercial Unix world is minimized.
- Torvalds used from the beginning the ***Usenet*** to develop Linux. Thereby development was not connected to a physical location. This was a precondition for making distributed innovation work.
- When Torvalds started programming Linux the operating system ***Minx*** had a Usenet community of approximately 40'000 members. Back in the early nineties in which computers and online connections were rare this number was huge. Interested in advancing Minix its users proposed patches and new features. However, the owner of Minix Andy Tanenbaum did not accept these patches and new features because he wanted that Minix remains small so that it can be used as an educational operating system. Instead of being pleased receiving proposals for advancing the operating system Tanenbaum lamented. Not being able to contribute actively to Minx its users got frustrated. At the same time the Minix license did not allow „to do a lot" (Torvalds & Diamond 2001: 100). Torvalds

[220] Another reason for the success of Linux in contrast to the BSD family was a lawsuit that caused some concerns in the BSD community and motivated some developers to switch to Linux (Raymond 2004: 41).

[221] Tanenbaum (2000) relicensed Minix in 2000 under a BSD license. Until then it was under a proprietary license.

[222] Legally it is not required to give modified source code back to the community *as long as the software is not redistributed.*

> accepted in difference to Andy Tanenbaum patches and new features. In consequence, many developers switched from Minix to Linux because they were accepted as core developers. It seems to be that the responsive project management style of Torvalds on the one hand and a large community eager to contribute source code on the other hand were very important for the success of Linux.

The project structure and mechanisms of Linux clearly contributed to its success. There were often new releases if patches were merged. This was done even if the source code was not perfect which kept development moving on. Communication in Linux development is open and accessible by everyone. In contrast to this style discussion in the FreeBSD project was hierarchical and not everyone could join mailing lists.

Linus Torvalds' development style not only activated software developer to contribute to given problems. With the time going on this style got a self-runner. Torvalds (2001: 90) reports that after having solved the most fundamental problems he lost interest in development. However, two things prevented that the project did not perish. *Firstly*, Torvalds destroyed his Minix partition by mistake so he was forced to use Linux further. *Secondly*, people kept sending feedback, patches, and new features. Linux developers expected Torvalds to review the contributions and merge them. In fact, while Linux is certainly a good product its initial success is perhaps based on management innovation (Hamel 2006: 74).

In consequence, the factors responsible for the success of Linux were the GNU GPL license, the incorporation of other programmer' input, as well as early and often releases.

4.2.2 NetBSD

After having tracked some factors contributing to the success of Linux another example is presented to show how governance structures and mechanisms may contribute to failure.

In August 2006 an email of Charles Hannum (2006), co-founder of the NetBSD project, attracted some attention. Hannum claimed that the NetBSD project has stagnated to the point of irrelevance. For Hannum (2006) there are mainly four problems related to the decline of the project:

- There was ***no strong leadership*** who set goals and directions and was able to get people to do what must be done.[223]
- There was some sort of ***"ownership" of code*** in the sense that developers were territorial about their code and did not encourage other developers to work on the same piece of source code. In consequence, projects were locked because a few developers did not move on. As a result development was slowed down. This is in contrast to the Linux kernel. If somebody does not produce source code somebody else will.
- ***Public code review*** in the NetBSD has gotten very poor. A major architectural change was for example developed entirely in secret without public review.
- Also part of the problem was the ***commercial firm*** Wasabi that entered into the NetBSD hierarchy. This resulted in "some very strange and discouraging actions" (Biancuzzi 2006). For example a NetBSD release was held up for more than a week only to do a combined NetBSD and Wasabi press

[223] By referring to the OpenBSD project Hannum states that leadership may be not only too weak but also too strong (Biancuzzi 2006).

release. The developers except the employees of Wasabi (e.g. the release manager) did not know about the corresponding secret arrangement. People with interest in Wasabi also wrote new bylaws in secret and after presentation to the community every input was discarded or ignored. Furthermore, Wasabi poached NetBSD hackers who afterwards stopped contributing to the NetBSD project. At the same time Wasabi did not contribute significantly to the NetBSD project. As Hannum concludes Wasabi can effectively do whatever they want and there is no check and balance.

- The NetBSD Foundation had different ***inefficient executive committees*** "which mostly do nothing, … they serve only to obfuscate things". Hannum claims that the NetBSD Project should have been "managed based on technical merits." Hannum argues further that "there must be no perceived glamour in participating in the Foundation; it must be composed of people doing it because they are dedicated and want to help the project."

The problems of the NetBSD project seem to be rooted in its development processes. Public peer review and open development is a key element of FOSS projects. At the same time a commercial firm got very influential and decisions were based neither on technological reasoning nor on democratic procedures.[224] It seems to be obvious that developers, especially those motivated by the joy of programming, will leave such a project.

Based on the successful example Linux and the less successful example NetBSD it is suggested that certain aspects of the FOSS production model respectively of its governance structures and mechanisms are important for project success. In the following the FOSS production model will be discussed to find out more why technically successful projects such as FreeBSD and NetBSD "never caught fire the way Linux has" (Raymond 2003, Linux).

4.3 FOSS governance structures and mechanisms

There are different descriptions of the FOSS development model. Probably the most prominent description of the FOSS development model is Raymond's (1999a: 27-78) comparison between the bazaar model in FOSS development and the cathedral model in traditional software development. In the cathedral model tasks and roles are clearly defined (Robles 2004: 194-195) and development is highly centralized. In the bazaar model roles are not properly defined and vary. There is a permanent interchange of knowledge and a flow of ideas that emerges spontaneously. Robles (2004: 194-197) summarize the bazaar model in the following:

- Treat your users as co-developers
- Release early
- Frequent integration (release often)
- Maintain several versions
- Modularize to the maximum
- Dynamic decision making structure

Feller and Fitzgerald (2002: 84) suggest that the following seven characteristics are relevant in FOSS development:

[224] In many FOSS projects non-democratic decision making is accepted as long the decisions are based on technological considerations and the developers trust the leaders.

- Parallel rather than linear development
- Involvement of large communities of globally distributed developers
- Utilization of independent peer review
- Provision of feedback to user and developer contributions
- Inclusion of highly talented and highly motivated developers *by self selection* (Feller & Fitzgerald 2002: 24-25)
- Rapid release schedules
- Increased levels of user involvement

Some of the mentioned characteristics of FOSS development are either overly enthusiastic[225] (highly talented programmers) or less relevant *for this study* (parallel development). Below only characteristics that are important for the FOSS development model, to some extent relevant for this study, and realistic are discussed.

However, before discussing the different elements of FOSS development at length some important remarks have to be made:

- ***Firstly***, FOSS projects are defined by its licenses terms respectively the freedom granted to users and not by the development model. Therefore, a project following another development model than specified here may nevertheless be a FOSS project as long as the source code is under a corresponding license.
- ***Secondly***, most descriptions of the FOSS development model such as Raymond's (1999a: 27-78) bazaar describe fairly big project with at least moderate success. As discussed in chapter 2.4 most FOSS projects are small and many projects have only one developer. The mode of production in such projects does probably differ from the development model described below. If there is only one developer in a project important elements of the FOSS development model such as peer review are either not possible or do not make any sense.
- ***Thirdly***, there are some prominent FOSS projects started, managed, and controlled by commercial firms such as Sun (OpenOffice.org) or MySQL AB (MySQL, Robles 2004: 195). Projects maintained by commercial firms probably follow a more traditional software development model. Therefore, it can be concluded that such FOSS projects not necessarily adopt what is called here the FOSS development model (Robles 2004: 194).[226]
- ***Fourthly***, contingency theory (Blau 1963; Blau 1970; Lawrence & Lorsch 1967; Pugh & Hickson 1976; Pugh & Hinings 1976; Pugh & Payne 1977) teach us that there is no "one best way to organize" (Galbraith 1977: 28). Therefore, "the same ingredients that made another project successful" will fail if taken for a new FOSS project (Robles 2004: 196). Similarly, an appropriate strategy can only be selected depending on the environment (Miller & Friesen 1983a: 222). Adapted to FOSS development

[225] It is known that there are substantial variations in programmer performance (Curtis 1981; Curtis, Krasner, & Iscoe 1988: 1271). Curtis et al. (1988: 1271) suggest that individual performance is a combination of motivation, aptitude, and experience. It is not obvious why these factors contributing to performance should be very unevenly distributed between FOSS and commercial software projects.

[226] However, the freedoms granted by FLOSS licenses are in contrast a necessary condition for the FLOSS development model (Robles 2004: 194).

it can be assumed that appropriate governance structures and mechanism have to be selected depending on different factors such as the motivation of the involved developers or the type of software. Optimal governance structures and mechanism for projects dominated by reputation investors are probably different from projects dominated by pro-social developers. However, in either case the bazaar described below is hard to reproduce and subject to forces not completely understood yet.

- ***Fifthly,*** the pure existence of certain mechanisms or structures in FOSS development does not necessarily mean that they are *positively* related to success.[227] It may be that some of the observed characteristics inhibit project success.

The comments below are restricted to community sponsored projects which managed to reach a minimal size and which had minimal success. Bearing these reservations in mind the different characteristics of the FOSS development will be discussed below.

4.3.1 Peer review and egoless programming

Peer review is an essential part of the FOSS development process (Feller & Fitzgerald 2002: 24). In fact, "massive independent peer review of code" is a "principle theme" in FOSS production (Bollinger, Raymond, & Trader 1999: 86). One of the ten big rules for committers in the FreeBSD project is to "discuss any significant change before committing" (FreeBSD 2005). It is asked to submit changes to the mailing lists. As the FreeBSD committers guide specifies further "this does not mean that you have to ask permission before correcting every obvious syntax error or manual page misspelling, simply that you should try to develop a feel for when a proposed change is not quite such a no-brainer and requires some feedback first." However, the FreeBSD guide concludes: "When in doubt, ask for review".

In software development peer review is not limited to FOSS projects. In the early seventies Gerald Weinberg (1971) wrote an influential book on the psychology of programming. One of his arguments was coined "egoless programming".[228] This mode of production is very similar to FOSS production: source code is reviewed in teams (Weinberg 1999: 119). Following this advice developers in egoless programming should not be territorial about their code and encourage other developers to look for bugs and potential improvements. As Kernighan and Plauger (1976: 2) put it:

> *…egoless programming means letting other people read your programs, without feeling wounded when they improve them.*

Raymond (1999a: 61) considers egoless programming as a vital correction of Brooks' Law (Brooks 1995: 25) that was also heavily influential in software project management. In Brooks' (1995: 13-26) book "The mythical man-month" it is argued that the complexity and communication costs of a software project rise with the

[227] For a discussion about measuring the success of FOSS projects see Crowston (2006).

[228] Raymond (1999a: 61-62) argues that although Weinberg's arguments are very compelling, they probably did not gain the merited acceptance because of naming the observed phenomenon as egoless. In a reprint of Weinberg's argument more than two decades after the first presentation Weinberg (1999: 120) asserts that probably he should have written "less-ego programming" to prevent unnecessary discussion on terminology.

square of the number of developers while productivity only rises linearly. [229] Raymond argues that if this were the whole picture Linux with its bulk of developers would hardly be possible.

The rationale behind peer review is apparent in Linus' law that states "given enough eyeballs, all bugs are shallow" (Raymond 1999a: 41). This signifies that if there are enough reviewers within the project most problems will be found and solved. Peer review is first of all a quality management system in which produced source code is examined by other FOSS project members. In the FOSS development model this peer review mechanism to filter out poor or even evil contributions is essential (Benkler 2002: 376). Peer review differs from management review or software audits insofar that the reviewers are project members writing source code of their own. The reviewers are neither part of the management nor external programmers. Weinberg (1999: 119) predicts that with egoless programming more errors can be found. A programmer who has written the source code originally internalized it and reached a point where he cannot see the wood for the trees. In consequence, he misses major errors. Besides of being part of the quality management system peer review has further advantages:

- ***Understandable source code:*** Because programmers know that others will review it they write the source code in a style understandable by others (Weinberg 1999: 120). Public discussion of source code may be a motivation not to contribute ugly code[230] and produces peer pressure.[231] As a developer in the FreeBSD study comments "embarrassment is a powerful thing" (Jorgensen 2001: 329). Source code written in a good programming style has fewer errors, is easier to debug, and easier to modify (Kernighan & Pike 1999: x). This makes the organization additionally less dependable on the original programmer if the software is developed further. It is sometimes thought that in contrast to FOSS projects in-house development of commercial firms tends to produce spaghetti-code (e.g. Young & Rohm 2000: 147).[232]
- ***Feedback:*** Peer review provides concrete feedback for the producer of source code. This may be important for enhancing learning and human capital. Additionally, it is motivating that the own source code matters and that one is a member of a social group (Torvalds & Diamond 2001: 111). In the FreeBSD study 32% of the developers got feedback in the form of comments on their last source code contribution from three or more persons, 33% got feedback from two persons, 21% from one

[229] The often cited sound bite of Brooks (1995: 25) is that "adding manpower to a *late* {emphasis added} software project makes it later". This is probably even true if applied to FOSS projects. However, the reasoning behind the statement that costs rise with the square of the number of developers while productivity only rises linearly can be questioned in the domain of FOSS production. Brooks argumentation only applies if developers are added *late*. Many FOSS projects have no strict schedule and "late" and "later" do therefore not apply.

[230] Interestingly, in the FOSS and UNIX community writing beautiful code and having a good taste is very important (see for example Levy 1984: 43; Thompson 1984). For example Ken Thompson (1984: 761) addressed in his Turing Award Lecture (the Turing Award is thought to be the Nobel Prize of computing, Gold 2004) not Unix itself but the cutest program he has ever written. An outsider does probably not pull beauty and programming together. Gelernter (1999: 22) argues that beauty is more important in computing than everywhere else in technology because of its high complexity. In fact, writing beautiful source code is for Gelernter a mechanism to deal with complexity. Beauty is defined as the quality of being pleasing to the senses or the mind (Hornby 2000: beauty). The urge to write beautiful source code may be explained by referring to tacit knowledge that may mastering complexity better than explicit knowledge.

[231] In FOSS development not only peer pressure may lead to the contribution of proper source code but also voluntary self-commitment (for enforcement of self-commitment see Schelling 1985).

[232] Spaghetti code is a pejorative term for source code with complex and tangled control structures (Raymond 2003: spaghetti code).

person, 11% got no feedback, and 3% never distributed source code to receive comments prior to committing (Jorgensen 2000, 2001). Within a month 44% of the developers received a bug report and 43% a bugfix.

- ***Project information:*** Discussion on contributed source code is not only very insightful for contributors themselves but also for lurkers who do not or not yet contribute to the project. All programmers participating or following a peer review discussion learn from each others inputs that raises the overall competence of all project members.
- ***Project documentation:*** Discussions in mailing lists represent technical debates. It produces then at the same time as a byproduct a project documentation (Berglund & Priestly 2001: 133) and technical guidelines how to write source code. For example heavy critique about a certain naming of identifiers shows prospective contributors how to do it better.
- ***Basis for decision making:*** The result of peer review serves as a basis for decision making for FOSS project leader. It informs whether a certain piece of source code should be merged or not.
- ***Planning instrument:*** Last but not least discussion on contributed source code is sometimes not only a technical evaluation but also a way for communication (Curtis, Krasner, & Iscoe 1988: 1280). Reviewed source code may be technically sound but does not fit with the intended direction of project development. Peer review discussion serves then as way for determining future development. Peer review may therefore also be a planning instrument. The advantage of such a planning process is that it is not solely based on theoretical reflections but also on existing source code. Communication channels such as mailing lists are "a court of public opinion" (O'Mahony 2003: 1189).

Weinberg comes to the conclusion that "there is now no shortage of evidence that … technical reviews lead to more reliable code produced more cheaply and consistently" (Weinberg 1999: 120).

There is also peer review in conventional software development. However, in FOSS projects it is in many cases adopted more consequently. Additionally, in contrast to commercial projects peer review in FOSS projects is mostly organized informally. Generic formal processes as defined in the corresponding IEEE standards are not common (IEEE Standards Board 1989). Feller and Fitzgerald (2002: 88) argue further that there is an immanent factor that restricts the functioning of peer review in commercial but not in FOSS projects. In commercial projects peers may be dependant on each other's goodwill for career advancement and job security and therefore will not be motivated to dig deep. At the same time one may assume on the contrary that peers in commercial settings are overly critical due to organizational politics. Of course, to go easy on with peers or hypercritical comments do also exist in FOSS projects. Yet, it can be assumed that in FOSS projects these mechanisms are less important because developers are by far less dependent on each other (see below). Project members can leave the project at any time and fork it if they do not agree with a certain decision.

Weinberg (1999: 119) argues that the reason why not all commercial firms adopted egoless programming is that knowledge is seen as valuable proprietary information that should be kept secret. In consequence, the source code is shown to as few programmers as possible. This contradicts the concept of peer review in which source code should be reviewed by as many as possible.

Similar to FOSS development peer review is also important in the scientific development model (Benkler 2002: 381; Bezroukov 1999a). Science is comparable to FOSS development because both are coordinated in a spontaneous way of independent initiatives (Polanyi 2000: 2). Polanyi (2000: 3-4) argues that science advances most if every scientist contributes what he think he is personally capable. The rationale behind this is that the scientist himself knows best what his strengths are. An attempt to organize the scientists under a single authority would eliminate their independent initiatives and reduce thereby their joint effectiveness.

By referring to the academia Bezroukov (1999b) presents a skeptical view of peer review and egoless programming. He argues that in academy simple, objective, and effective peer review is only a statistical approximation. Bezroukov states that similar to academic peer review in FOSS development "your work can be rejected out of overload, petty jealousy, incompetence or political motives" and "for similar reasons, the concept egoless programming does not really work in open source". Bezroukov argues further that similar to FOSS development "scientists are eager to share their discoveries" but that "they usually carefully guard and often conceal the exact methods that lead to those discoveries". Bezroukov's objection has to be taken seriously. Peer review is no silver bullet. However, Bezroukov's critique on peer review is based on a comparison with an ideal world in which the former is clearly inferior. If peer review is compared to other types of quality management the assessment of peer review may come to a far better conclusion than the one of Bezroukov. Additionally, there is a very powerful mechanism in FOSS development that protects against unfair peer review: Forking (see chapter 2.7). If a community does not accept a feature or a bug fix developers may fork the project and merge the corresponding source code themselves.

4.3.2 Recruitment of project members

FOSS developers who are not paid choose the problem they want to solve themselves. It is not the projects that find developers but developers who find projects. Moody (1997) suggests that the "automatic selection of programmers to work in the areas they know best" is a characteristic of FOSS development. On a question how the FOSS community ensures that all necessary development problems get solved Raymond answered (Bollinger, Raymond, & Trader 1999: 88):

> *By letting people choose their own problems. There are enough people that eventually all the significant problems get solved – it's kind of a Chinese-army approach. Instead of taking a small number of people and saying, "OK, we say you're going to go here, because that's what our priorities are," you just create a framework. Within that framework, anybody who wants to solve a problem can bootstrap his own project and collect a community around it, and then use that community to solve the problem. You gain two things that way. One, you don't have to spend management resources trying to coerce or bribe people into doing what they really don't want to do. Two, you end up with work done by the people whom you most want doing it – the most motivated ones.*

In contrast to commercial firms the relationship in the FOSS community between developers and project organizations is not regulated by contracts (Dahlander & Magnusson 2005: 484).[233] FOSS developers are free to join every project they are interested in. Indeed, the self-selection of developers seems to be a very important precondition for maintaining motivation (see also Feller & Fitzgerald 2002: 24-25). Additionally, a side effect of self-selection is that developers make their contributions in an area that is appropriate to their

[233] Exceptions are FOSS developers paid by commercial firms. However, Red Hat, a commercial Linux distributor, pays payrolls for Linux developers without having ever taken influence on what they actually work (Young & Rohm 2000: 167).

skill level. Linus Torvalds (2001: 121) argues that the “most effective way to lead is by letting people do things because they want do them, no because you want them to.”

In an interesting paper Benkler (2002: 374) focuses on the specific mode of FOSS organization respectively development. He is interested in the dynamics that makes FOSS possible and successful (Benkler 2002: 375). Benkler argues that all collaborative production systems have an information problem. Agents in such systems need to know what they should do. Markets solve this problem by price signals that are attached to certain alternative courses of action. In firms this problem is solved by different signals from different agents that have different weight. In short, what hierarchically higher supervisors say counts. Both the market as well as the hierarchical solution have the characteristic that the objects of the signals have to be specified. If such objects cannot be specified they cannot be properly priced or managed. In the commons based peer production model the information problem respectively the problem for identifying and allocating human creativity to work is solved differently. Individuals self identify the task the want to perform (Benkler 2002: 376). The point of decision about assigning manpower to any given set of resources is then within the individual itself. This is an advantage because individuals themselves have mostly the best information about their own ability to solve a certain task. This mechanism of self-allocation is especially beneficial if there is high variability among individuals and if the resources are difficult to specify from the outside. Motivation and creativity are certainly resources with high variability and that are difficult to specify. Brooks (1987: 18) notes that software development is a creative process and that people are at the center for improving software. Benkler (2002: 377) argues in consequence that the advantage of peer production such as in the FOSS development model is the identification and allocation of human creativity. He assumes that the peer production model gets more important because human creativity gets increasingly crucial.

The downside of the described recruitment mechanism respectively self-selection is that planning gets difficult (Robles 2004: 199-200). In the traditional software development model one knows more or less anytime the human resources assigned to a certain task. As Torvalds put, Linux has “random people in random countries working on random things” (Hamm 2004). It also may be that tasks that are neither fun to do nor give reputation move only slowly.

4.3.3 Computer-mediated interaction

Interaction is the essence of any interpersonal relationship (Thibaut & Kelley 1959: 10). An integral element of FOSS production is that interpersonal interaction is computer-mediated and that collaboration is organized online (Feller & Fitzgerald 2002: 24 and 125-129).[234] For most projects there is no or little face-to-face communication (Feller & Fitzgerald 2002: 95; Mockus, Fielding, & Herbsleb 2002: 310), which may have some implications.[235] It is thought that signs like facial expression or voice tone reveal intentions (Frank 1993: 165). Computer-mediated online collaboration has the consequence that emotions of other subject cannot be directly observed and that therefore emotions cannot be used for the prediction of behavior (Orbell & Dawes 1991: 516). However, trust can be build up despite the lack of face-to-face communication (Wilson, Straus, &

[234] The only notable software programmer known without email is Donald Knuth (2006). He used emails from 1975 until 1990. He argues that 15 years of email use is plenty for one lifetime.

[235] The collaboration of geographically distributed teams is not a phenomenon restricted to online collaboration. The Hudson’s Bay Company (Kuster 2006) as well as the Roman Empire (Brytting 1986) are examples of collaboration without face-to-face communication from the past.

McEvily 2006: 29). The downside of building up trust online is that it needs more time. Additionally, communication frequency decreases with distance but after approximately 30 meters communication frequency remains constant (Herbsleb et al. 2001: 320). Therefore, communication intensity is higher with cubicle neighbors but this effect disappears very soon and it does not depend whether a coworker is located 100 meters away or at another continent.

In an experiment of Hiltz et al. (1986) on differences between computer-mediated and face-to-face communication the former was more task oriented and less oriented towards social-emotional functions. The equality of participation was in groups with face-to-face communication lower than in computer-mediated groups although this effect was not significant. Straus (1997: 255) replicated the observation that in experiments computer-mediated teams are more task oriented and that there is more equal participation in discussion. In yet another experiment there was no difference concerning the equality of participation (Adrianson & Hjelmquist 1999: 186-187). A possible explanation that subjects in computer-mediated communication are probably more uninhibited (Kiesler et al. 1985: 98) is the greater anonymity which generates more original solution than in non-anonymous communication (Connolly, Jessup, & Valacich 1990).

Adranson and Hjelmquist (1999) conducted experiments to research the influence of the communication medium on social dilemmas and criminal puzzles. Although communication in computer-mediated and face-to-face teams showed different patterns this did not have consequences for the final outcome. They concluded that the structure of the problem (social dilemmas vs. criminal puzzles) is more important to the final outcome than the communication medium (Adrianson & Hjelmquist 1999: 192).

From the results presented above it can be concluded that the channels of communication do not influence behavior sustainable.

4.3.4 Release early and often

According to Raymond an important factor for the success of Linux are the management capabilities of Linus Torvalds (Bollinger, Raymond, & Trader 1999: 89). Prominently, he released a new kernel every 24 hours that was thought to be bizarre in operating system development. Nowadays, a rapid and incremental release schedule is characteristically for most FOSS projects (Feller & Fitzgerald 2002: 24).[236]

To release early has the advantage that the possibility of finding additional developers increases (Robles 2004: 196). In the FOSS community prospective contributors want to see some source code before they participate. According to Raymond (1999a: 58) there is a need to present developers a plausible promise. The source code can be crude, buggy, incomplete, and poorly documented but it must run and convince developers that there is a chance for success.

Another advantage of releasing early and often is that it may be a source of motivation to see that the efforts actually pay off. Writing a big masterpiece (such as a dissertation) without knowing whether it is successful may be frustrating (Torvalds & Diamond 2001: 82). In the FreeBSD study a developer states that there is a tremendous sense of satisfaction working in a “see bug, fix bug, see bug fix get incorporated” cycle (Jorgensen 2005: 227). Brooks (1987: 18) argues that enthusiasm jumps when there is a running system, even it is very simple.

[236] Interestingly, Polanyi (2000: 3) argues that science can only advance incrementally because the end-result of scientific discovery is unknown.

To release early and often is heavily related to incremental development. Brooks (1987: 18) notes that nothing has more radically changed his effectiveness to code than incremental development. The huge advantage of incremental development is how it masters complexity. Brooks argues that software is to complex to specify it properly in advance. In fact, Linux was not an intentional masterstroke but an incremental process in which experiments, ideas, and tiny scraps of code gradually came together into an organic whole (Moody 1997).

Luethi (2004) argues that software production started historically as a very pronounced high cost situation that required large investments before the first lines of source code were written. However, the "release often and early" principle of FOSS development fostered incremental contribution that in turn resulted in software production as a low cost situation. Because of incremental development even small and therefore cheap contributions got possible. Furthermore, if small contributions are possible it is more likely that there is a privileged user who has an incentive to pay all the costs for the public good on his own (see also chapter 5.5.1.2). A user may contribute only one bug fix that he desperately needs but nothing else. For doing so he does neither need to apply for formal project membership nor does he has to sign a labor contract.

4.3.5 User involvement

A critical success factor of software projects is to understand users (Reel 1999: 19). Furthermore, unlike writing a novel creating good software must include collaboration with its users (Messerschmitt & Szyperski 2003: 2 and 83-84). In his Turing Award Lecture[237] Dennis Ritchie (1984: 759) argued that a very important factor for the success of Unix was the fact that users were at the same time developers. In FOSS projects the developer-user collaboration is at the heart of the development model (Feller & Fitzgerald 2002: 92).

The incorporation of users for software improvement is important because not all bugs are easy to detect by developers.[238] Some bugs can only be detected by users that run software for a long time and in an unanticipated manner.[239/240] An example is the software bugs in the Therac-25 radiation therapy machine that caused the death of several patients. Reports of users respectively physicians about radiation poisoning were

---

[237] See footnote 230.

[238] Such a hard to detect software bug in the energy management system contributed heavily to the power outage all over North America at August 14, 2003 (Poulsen 2004). This bug never turned up until the blackout and was hard to find even after the incident. Because of the bug the alarm system failed to work and retarded the response that could have prevented the blackout. Other major causes were inadequate system understanding, inadequate situation awareness, inadequate tree trimming, and inadequate diagnostic support (U.S.-Canada Power System Outage Task Force 2004: 18-19).

[239] However, it is not possible to ship software that does not have any bugs (Sink 2006). Furthermore, to fix bugs also have disadvantages. Doing so may be problematic because code changes are risky. Every time a bug is fixed a new and perhaps even more severe bug may be introduced.

[240] An example of a software bug that manifested itself only after an extended time of running without reboot was the malfunction of a Patriot missile-defense system in which 28 U.S. Army reservists were killed (Grottke & Trivedi 2007). The software of the Patriot system needs time as a real value to project a target's trajectory. However, the operating system kept time internally as an integer, counting tenths of seconds and storing them in a 24-bit register. Converting it into a real value caused an imprecision. The real value cannot be more precise than 24 bits. The imprecision is proportional to the length of time the system had been continuously running. Although this was known it was not assumed that Patriot systems are operated so long that a failure becomes ever imminent. In fact, the Patriot system was designed to be mobile and operate for only a few hours at one location to avoid detection (General Accounting Office 1992). The Patriot battery in question had been in operation for over 100 consecutive hours that resulted in an inaccuracy of 0.3433 seconds. The problem could have been easily solved by rebooting the system every few hours that takes about 60 to 90 seconds. Rebooting reinitializes the computer's clock. However, one may assume that the soldiers were neither trained to reboot the system nor did they understood how time is computed by the operating system.

first attributed to a failure in the microswitch (Leveson & Turner 1993: 23). After the machines were technically adjusted to such failures a radioactive contamination was thought to be impossible because safety has thereby improved by "10'000'000 percent" (Leveson & Turner 1993: 27). A very big problem was that the manufacturer was never able to reproduce the error. Finally, a physicist started his own investigation and found that "data-entry speed during editing was the key factor in producing the error condition: If the prescription data was edited at a fast pace (as is natural for someone who has repeated the procedure a large number of times), the overdose occurred" (Leveson & Turner 1993: 28). The employees of the manufacturer who tested the machine were slower in data-entry than the operators in the hospitals. That is why they could not reproduce the error and in turn thought that there is no malfunction.

Within one month 44% of developers in the FreeBSD project received at least one bug report and 43% received at least one bug fix for the source code they are responsible (Jorgensen 2000, 2001). Users are especially valuable to find Heisenbugs. Gray (1985: 17-19) coined the distinction between Heisenbugs and Bohrbugs. Heisenbugs disappear or alter if they are researched by developers (Bourne 2004: 32).[241] For example it may be that bugs do not manifest if a debugger is used. For developers it is difficult to find bugs if they are related to strange hardware conditions (rare or transient device fault), limit conditions (out of storage, counter overflow, lost interrupt), or race conditions (Gray 1985: 18). In contrast to a Heisenbug a Bohrbugs does not change if researched and "manifests reliably under a possibly unknown but well-defined set of conditions" (Raymond 2003, Bohrbug). The importance of users reporting Heisenbugs is supported by Gray's (1985: 18) argument that Bohrbugs respectively bugs that always fail on retry are mostly gone in mature software. In an experiment on software failure Gray (1985: 18-19) found that one out of 132 bugs was a Bohrbug and the rest were Heisenbugs.[242]

In the following statement Microsoft (2005b) refers probably to user involvement:

> *Microsoft has been learning from the OSS community regarding the benefits of deeper collaboration and increased transparency leading to better communication with customers.*

Kyng (1991: 66) emphasis similarly the importance of cooperation between users and developers as well as mutual learning in commercial software development. This level of user involvement contradicts the traditional waterfall model in which the user is primarily involved in the requirements specification.

An example what may happen if software developers do not respond to users is apparently the Minx-Linux story. Minix is an operating system created in the late eighties by Andrew Tanenbaum for educational purposes. In 1987, two months after the Minix release, a newsgroup[243] with a community of 40'000 users was established (Moody 1997). Back in the late eighties computer usage and online communication was not common which makes the Minix user base very impressive. Tanenbaum reports that he "was getting hundreds of emails a day asking to add this feature and that feature". He constantly refused to do so. In consequence,

[241] Other classes of bugs like Schroedinbugs are also hard to find for developers. Schroedinbugs manifests themselves only after the software is used in an unusual way that is not anticipated by the developer (Raymond 2003, schroedinbug). Software with a Schroedinbugs can run for a long time without any problem and then suddenly and completely stop working.

[242] The methodology for finding Heisenbugs was as follows (Gray 1985: 18): „The spooler error log of several dozen systems was examined. The spooler is constructed as a collection of fail-fast processes. When one of the processes detects a fault, it stops and lets its brother continue the operation. The brother does a software retry. If the brother also fails, then the bug is a Bohrbug rather than a Heisenbug."

[243] Comp.os.minix

people got frustrated. At this point Linus Torvalds appeared in the Minix newsgroup with his announcement of Linux. Linus Torvalds responded to bug fixes and suggestions for adding functionalities. As a result developers left the Minix community and joined the Linux community.

Drucker (2001) argues that a main characteristic of the knowledge society is upward mobility available to everyone. The reason for this is that knowledge can be acquired by everyone. Similarly, in software development the foundation of user involvement is the freedom to run, copy, distribute, study, change, and improve the source code. Neither the freedom to run nor the freedom to study the program such as specified in the Microsoft Reference License is sufficient for user involvement.[244] For many user involvement is meaningless if he is not allowed to change and redistribute software.

User involvement is accompanied by individual role changes. Several individuals in the FOSS community shift between a user, a developer, and an owner role (Feller & Fitzgerald 2002: 111). This may also explain why FOSS is, at least until now, more successful for the back-end than for the front-end. Hackers work more with the command line while dummy user work with graphical user interfaces. For front-end tasks there are in consequence fewer developers who are also users. At the same time dummy users who utilize front-end tools are not capable to participate in software development.

4.3.6 Modularization

Although Simon (1981: 200-202) did not use the word modularity in his writings he was central to the theory of modularization. Simon narrates a parable of two watchmakers called Tempus and Hora. Tempus constructs watches so that if he puts the watch down to answer a phone call the watch fells to pieces and has to be reassembled from scratch. Hora produces his watches in stable subassemblies of ten elements. If Hora has to answer a phone call he loses only a small part of his work. By modularizing his work Hora is most probably more efficient.[245]

The concept of modularity was practically first used in the Unix world (Raymond 2004: 83-84).[246] Modularization is deeply rooted in Unix philosophy (McIlroy, Pinson, & Tague 1978, cited in Raymond 2004:12):

> *Make each program do one thing well. To do a new job, build afresh rather complicate old programs by adding new features. Expect the output of every program to become the input to another, as yet unknown, program.*

Software modules are small pieces of source code that can be conquered separately such as subroutines, functions, and procedures (Kernighan & Plauger 1978: 59). Kernighan and Plauger (1978: 160) argue that the coupling between modules should be visible and every module should hide something (Messerschmitt &

[244] The Microsoft Reference License is part of Microsoft's Shared Source Program. There are different licenses associated with the Shared Source Program (Microsoft 2005c). Licenses such as the Microsoft Reference License allow to view the source code but not to modify it. The Microsoft Permissive License (Ms-PL) "allows to view, modify, and redistribute the source code for either commercial or non-commercial purposes" and is probably a free software license. While ifrOSS (2006) recognizes the Ms-PL as a weak copyleft license the Free Software Foundation Europe (2005) still reviews it but assumes that it is a variation of the copyleft idea and that it qualifies as a free software license. It must be noted that not the publication of a license text is the critical move but whether relevant software is actually released under this license.

[245] However, it also may be that the costs for modularization are higher than the costs to reassemble a watch from scratch.

[246] For Baldwin and Clark (2000: 6) the first truly modular design in the computer industry was IBM's System /360.

Szyperski 2003: 87-90).[247] The modules should be connected by well defined interfaces (Raymond 2004: 84). Closely related to the establishment of modules in the Unix world is the invention of pipes. Pipes are a set of chained processes in which the output of each process is the input of the next process.

Modularization is not only important in the Unix environment but also in FOSS development (Feller & Fitzgerald 2002: 78; Narduzzo & Rossi 2003: 3).[248] Moody (1997) compares Linux, initiated by the Fin Linus Torvalds, with the Kalevala, the Finish epic poem. The Kalevala has 22'795 verses compiled by Elias Lönnrot taken from folklores that evolved over 2500 to 3000 years. Similarly, the Linux kernel is a source code patchwork. The architecture of the Apache HTTP server 2.2 is based on 81 different modules that come as part of the Apache distribution (Apache Software Foundation 2006a). Each module handles different functionalities and each module can be individually modified. An even more extreme example of modularity in the FOSS world is the programming language Perl that consists of more than 10'000 modules (CPAN 2006).

Modularization has some important advantages:

- ***Code reuse:*** Well designed modules can be reused in other applications (Kernighan & Plauger 1978: 59). In contrast to witchcraft in which the actors have secrets and guards it is a FOSS principle that developers look at other developer's source code and build their own contribution on top of them (Hamm 2004).
- ***Different programming languages:*** Modules can theoretically be written in different programming languages and it is possible to compile them independently (Kernighan & Plauger 1978: 59). To do so may be necessary if source code is reused. Additionally, different languages have different strength and weaknesses. Therefore, theoretically different programming languages could be used for different tasks although this complicates designing the interface.
- ***Complexity reduction:*** Modules add a structure to the source code and reduce thereby complexity that is very important for programming (Kernighan & Plauger 1978: 95). According to Kernighan (1978: 59) modularization is the only way to read and understand programs. It also allows programmers to write a module with little knowledge of the source code in other modules (Parnas 1972: 1053) and therefore simplifies development (De Goyeneche & De Sousa 1999: 65). Coordination between modules and loosely coupled groups is not needed *apart* from designing interfaces. Modularization also simplifies bug fixing because it narrows the source code in question.
- ***Prevention of strange interaction:*** Modularization prevents "things from interacting with each other in unexpected ways" (Kernighan & Plauger 1978: 95).
- ***Parallel development:*** Modularization enables parallel development that in turns allows many developers to work simultaneously on a project that shortens development time (De Goyeneche & De Sousa 1999: 65). Modularization requires no coordination apart from designing interfaces. Functionalities can be extended in a separate module without changing the core functionality (O'Reilly 1999: 37).

---

[247] For an explanation why information hiding is important see Parnas (1972).

[248] In a comparison of Linux, the Apache HTTP server, and the GNU Compiler Collection with three different closed source wireless protocol products Paulson et al. (2004: 253-254) found no indication that FOSS projects are more modular than closed source projects. However, the selection of projects is highly unrepresentative. It may be that software in certain areas are more eligible for modularization than in other areas irrespective whether it is a open or closed source project.

- ***Efficient upgrading:*** The replacement of modules is possible without reassembling the whole system (Parnas 1972: 1053). The modularization of the Linux kernel makes it possible to add code dynamically to the kernel at runtime without rebooting (De Goyeneche & De Sousa 1999: 66).
- ***Bite-sized tasks:***[249] Modularization breaks a project up into small tasks. Each task can be handled separately in a shorter amount of time. In consequence, the motivation needed for contribution can be smaller (Benkler 2002: 378-379). If the modules are heterogeneous in size people with different levels of motivation may produce smaller or bigger contributions according to their level of motivation.

However, modularization is never perfect (Herbsleb & Grinter 1999: 10). There are also limits to modularization. An analysis of Hatton (1997: 96) showed that the relationship between fault density and module size is U-shaped. Small and big modules do have relatively more bugs than medium sized modules (200-400 logical lines of code).

Additionally, modules interfaces have to be well designed (Raymond 2004: 84). A related problem of modules is that sometimes interdependencies emerge which were at first not considered (Narduzzo & Rossi 2003: 7). Designing interfaces and integrating modules is therefore not trivial.[250] Benkler (2002: 379) also mentions that modularization needs a low cost mechanism for integrating the different modules into a product.

4.3.7 Decision making structure

Leadership may be defined as the process that influences the interpretation of events, the choice of objectives and strategies, the organization of work activities, and the like (adapted from Yukl 1998: 5). There are many different forms of leadership such as participative leadership, charismatic leadership, transformational leadership, or self-managed groups (Yukl 1998). In the case of self-managed groups responsibility and authority is in the hand of groups (Yukl 1998: 359). In a literature review Pearce and Ravlin (1987: 761) found that in self-managed work groups certain outcome values such as employee satisfaction are higher. Additionally, production costs decrease because of higher group member's innovation and lower absenteeism, turnover, and accident rates.[251] The influence of self-managed work group on productivity is likely positive although research has not yet come to a final decision.

In a study of FOSS projects de Souza et al. (2005: 201) analyzed network relationships between developers. They researched each an example of:

- a centralized project in which development is centered around a sole developer,
- a project with a core of seven developers which are strongly connected with each other while a periphery of four developers is not strongly connected, and
- a densely network of six interconnected developers in which each developer depends on the other developers.

Similarly to these different network structures there are projects in which a single developer has more influence in decision making than the rest of the project members and projects in which all developers are more or less equally important in decision making. Basically, in FOSS development there are two different

[249] The wording "bite-sized tasks" is adopted from the Subversion project (Subversion 2006).
[250] See for example the problems of the GNU HURD kernel and the GNU Linux approach (Narduzzo & Rossi 2003: 23-29).
[251] See also Banker et al. (1996), Cordery, Mueller and Smith (1991), Pearson (1992), and Wall et al. (1986).

types of leadership respectively decision making structure. In the *first type* a *benevolent dictator* rules the project. Very often this position is hold by the project founder. The *second type* is based on *democratic decision making*. In the following, these two types are first discussed and then compared.

The benevolent dictator

Linux is an example of a project with a benevolent dictator, namely Linus Torvalds, at the top (Hamm 2004). There are lieutenants below Torvalds deciding within their area of competence. Concerning benevolent dictators one should not think of a hierarchy in the classical sense. In the Linux kernel project Torvalds as well as his lieutenants are chosen by informal consent (Hamm 2004):

> *The lieutenants get picked. It's not me or any other leader who picks them. The programmers are very good at selecting leaders. There's no process for making somebody a lieutenant. But somebody who gets things done, shows good taste, and has good qualities -- people just start sending them suggestions and patches. I didn't design it this way. This happens because this is the way people work. It's very natural.*

Linus Torvalds notes on his role in Linux development (Hamm 2004):

> *What I do mostly is I'm a communications channel. I'm one of a couple of central points for discussions. I have all the patches come to me, though I have sub-lieutenants doing the programming work. I'm a meeting point, rather than a software engineer. I don't do much programming anymore.*
>
> *I don't decide what needs to be done. It's defined by what people need to get done and what they want to do. Getting it working together -- that's where I and other organizers come in. If I see something that needs more attention, I sometimes suggest something.*
>
> *If there's not enough effort going into a certain thing, it's usually because it's hard to get started on something new. Once somebody gets started, the others get into it. Occasionally I have to start a project and get it far enough along that it's self-sustaining, and then I pray for somebody to take over.*

According to Torvalds programmers follow him because they trust him (Hamm 2004). He states being a shepherd but not of sheep but of cats.

According to Taylor (1911: 28) the division of labor in scientific management should be based on “universally friendly cooperation between the management and the men” and not on “suspicious watchfulness” and “open warfare”. Similarly, this seems also to be an important precondition for successful FOSS projects. It is reported that Larry McVoy said that Torvalds is not the best software developer but good in motivating “losers” to bring out their best (Young & Rohm 2000: 177).

In fact, if developers are not happy with a benevolent dictator there is always the possibility to fork a FOSS project without loosing one line of code. Some of the BSD forks clearly show that this may happen if developers are not content with their leaders.[252] Weber (2004: 180) concludes that the right to fork code transfers a very important source of power from the leader to the followers. Every leader knows or should know that an unsatisfied developer may take the source code with him and start on it a competitive project. If enough other developers join the fork the original project will fail due to lack of human resources. A related example is the Mambo – Joomla! fork discussed in chapter 2.7. The entire team of core developer did not trust

[252] As Young and Rohm (2000: 169-171) report, there was also a time in which a Linux fork was possible.

the leaders anymore. In consequence, they forked the project and all core developer left the Mambo project to form a new project called Joomla!. FOSS development is in general inherently community driven and follows a grass-root approach (PITAC 2003: 7). If this is not acknowledged developers will leave and look for another project in which they can participate.

If a leader fails to satisfy developers an alternative to forking the project is to replace him. The maintainer of a certain module may be replaced if his contributions to the project are qualitatively or quantitatively not good enough anymore. As Bruno Haible, a contributor to the memory management of Linux, puts it: "When the main author doesn't improve his code anymore, other people will" (Moody 1997). In many projects developers are informally accepted as leaders but also informally dropped.

In a widely recognized typology of Raven and French (1960: 612-621; 1958: 83) five types for social power in small groups are identified:[253]

- ***Reward*** power is based on the perception that one may get rewarded.
- ***Coercive*** power is based on the perception that one may get punished.
- ***Legitimate*** power is based on the perception that an individual has a legitimate right to prescribe behavior.
- ***Referent*** power is based on identification with an individual.
- ***Expert*** power is based on the perception that an individual has some special knowledge or expertness.

The benevolent dictator of a FOSS community project can certainly neither rely on coercive power nor on reward power. His influence is based on referent power and mainly on expert power. It may be that in the cases in which the project leader is also the project founder also legitimate power may be a source of authority (see also below legitimate power based on voting results).

It can be summarized that benevolent dictators in FOSS development have to be highly considerate of the developers. Otherwise they will respond adequately.

Democratic decision making

The decision making procedures in the Apache Software Foundation are very different from the ones' in the Linux project. These procedures are much more formal and democratic. In general, decision making is driven by the people who in fact contribute to a certain Apache project. Apache cofounder Brian Behlendorf (Goldman & Gabriel 2005: i) compares FOSS decision making to a Quaker style consensus. Indeed, in the sphere of the Apache Software Foundation (2006c) it is thought that there should be consensus in decisions.[254]

[253] Pettigrew (1972) completes this typology by the power of control over information. However, this source of power does not seem to be relevant for FOSS development.

[254] Wicksell's (1896) unanimity rule for reliable political institutions has interesting implications for decision making (for competing interpretations of Wicksell's unanimity rule see Johnson 2005). The members of a political body draw up proposals for both public good production and taxes to finance it. In this situation politicians may be tempted to vote against the production of the public good to save tax money although each of them do like to see the public good produced. They may assume that others will vote for increased tax and that thereby the public good can be produced. In Wicksell's unanimity rule the political body vote for or against each pair of public good-tax proposal until one receives unanimous consent (Buchanan 1967: 116). Because of the unanimity rule any member of the political body risks losing the benefits from the expenditure if he vetoes the proposal. A politician will only vote against a proposal if

However, if consensus is not possible the decision is made by voting. The voting procedures itself are elaborated and depend on the topic (Apache Software Foundation 2006b, 2006c):

- For procedural issues the majority of votes decide. However, in special cases the concept of lazy consensus applies. If nobody objects within a certain time it is assumed that everybody accept the proposition.
- For code modifications there must be three positive votes and every single negative vote is a veto that cannot be overridden. However, a veto must be accompanied by a technical justification and show why the change is bad. A veto without justification does not count. Again, the lazy consensus may apply.
- For package releases the majority of votes decide if there are at least three positive votes.

A co-founder calls the Apache project a meritocracy: The more work you have done the more you are allowed to do (Fielding 1999: 43). The Apache Software Foundation (2006c)is made up by the following positions:

- A ***board of directors*** governs the whole foundation.
- The different ***project management committees*** govern and control the single projects entirely.
- ***Committers*** are developers who were given write access to the code repository and contribute to projects.
- ***Developers*** are users who contribute source code or documentation but who do not have access to the code repository. They actively participate in projects, mailing lists, and discussions by providing patches, documentation, suggestions, and criticism.
- ***Users*** provide feedback, report bugs, and suggest features.

Leadership based on meritocracy implies that power is not based on external administrative hierarchy (PITAC 2003: 7). This may make it difficult for commercial firms to adopt the FOSS development model.

Similarity between the two decision making types

Both for projects with a benevolent dictator and for projects with democratic decision making the importance of proved merit and talent are high. This can be exemplarily recognized by the mechanisms of giving access to the source repository in the FreeBSD project.[255] Approximately 200 developers have access to the source

his net benefits are negative (Buchanan 1967: 91). The solution of Wicksell therefore eliminates the dominant strategy to defect unilaterally. In fact, there is the possibility either to veto the proposal with the effect that a public good will not be produced at all or to accept it with the consequence that the public good will be produced completely and funded by everybody. Because it is in a social dilemma for all individuals better if everybody cooperates than if everybody defects (Dawes 1980: 169) the public good will be produced. Thereby the need for unanimity prevents free riding in decision making. Although the rule is to some extent appealing the problem is that for most questions there will be a single individual who does not consider the public good to be beneficial. An example is the Church of the Holy Sepulchre. Each the Greek Orthodox, the Armenian Apostolic, and the Roman Catholic churches have a veto right concerning common ground such as doors and *windows*. Because the three actors cannot come to terms with each other a wooden ladder located on a *window* ledge over the church's entrance remains there since more than a century. A photo of 1892 shows the ladder exactly in the same position as it is nowadays.

[255] Jorgensen (2005: 229) considers the FreeBSD organization being similar to Mintzberg’s (1985: 160-161) adhocracy. Characteristics of an adhocracy are a highly organic structure, power shifts to specialist teams, and little formalization

repository and therefore the authority to change the source code (Jorgensen 2001: 322). A developer states in the FreeBSD study (Jorgensen 2001: 329): “The way you get granted commit privileges is by first making enough code contributions or bug fixes that everyone agrees you should be given direct write access to the source tree”. The contributed source code should also be “of sufficiently impressive quality that no one objects and says, ‘Yuck, no, don’t give him commit privileges. He writes ugly code‘”. Meritocracy respectively this process of granting access to the source repository seems to be appreciated. In the FreeBSD study 57% of developers report that the ability to influence the development process is a major reason for participation in the project while only 28% report that it is a minor reason (Jorgensen 2000, 2001).

Benevolent dictator or democratic decision making?

There are no theoretical or empirical studies researching the question under which circumstances in FOSS development decision making is made or should be made by a benevolent dictator respectively by democratic processes.

However, in a study on worker ownership the author argues that collective decision making is rarely found if there is a high degree of heterogeneity in the workforce (Hansmann 1990: 1783). It seems that collective decision making does only function if everybody does about the same work and if everybody has roughly the same status. An example are professional service firms such as law firms in Switzerland in which all lawyers have roughly an equal education and each lawyer works more or less without substantial supervision. Ellickson (1993: 1349) argues that if group members have in contrast heterogeneous interests they are better of if they delegate power to a manager who may resist lobbying of the different members.

Clearly, leadership in FOSS projects is primarily the results of its historical roots. However, it would be interesting to research whether the degree of developer heterogeneity were to influence the leadership model. Following the arguments above FOSS projects with a high degree of heterogeneity should have a beneficial dictator while projects with a high degree of homogeneity should have a democratic decision making.

In several experiments White and Lippitt (1960) studied the effects of different social atmospheres respectively leadership behavior. The authors found that *autocratic leadership* produces a large amount of critical discontent and aggressive behavior. Not astonishingly, with this kind of leadership group action is dependent on the leader. In consequence, group conversation was less varied than for other leadership behavior. At the same time, the quantity of work done in autocracy was slightly higher than in democracy. *Democratic leadership* produced more friendly conversation, high work motivation, and in contrast to autocracy more originality. One may therefore argue that leadership behavior and the related social atmosphere will have an influence on important FOSS project characteristics. It may be that in consequence the leadership type is related to the motivation of FOSS developers. It could be assumed that hackers who are motivated by the fun of programming will ceteris paribus choose projects with a different leadership type than individuals investing in their reputation.

of behavior (see also Travica 1999: 6-10#). In an adhocracy work organization rests on specialized teams. The specialists form small project groups anew from task to task.

4.3.8 Code reuse

According to Raymond (1999a: 33) “good programmers know what to write. Great ones know what to rewrite (and reuse)”. There is no reason to write source code from scratch if it can be adapted, reused, or recycled (Raymond 2004: 9). The most obvious examples of code reuse are software libraries that are collections of subprograms. But also modules, algorithms, components, and single code fragments can be reused. Haefliger et al. (2007: 14) found for FOSS development “three broad forms of knowledge and code reuse in the core sample: algorithms and methods, single lines of code, and components”. The FOSS project AbiWord, a word processor, for example reuses source code from Libpng to display png graphics, from fribidi to display fonts, from zlib for compression, from LibXML for XML file parsing, from ispell for spell checking, and from libcurl to download files (Haefliger, von Krogh, & Spaeth 2007: 17).

Poulin (1997: 5-6) assumes that for *domain-independent* software (e.g. graphical user interfaces or math libraries) up to 20% and for *domain-specific* software (e.g. navigational aids for aircraft or financial service libraries) up to 85% of the source code can be reused. For *application-specific* software adjusted to a special customer requirement there is little code reuse potential. A NASA study on software reuse practices in the Earth science community found that within the last five years 79% of the subjects reused software that was developed outside their project (NASA Earth Science Data System Reuse Working Group 2006). Mili et al. (1995: 529) report different actual and estimated reuse rates between 15% and 85%.

The reuse of source code is of intense interest in software production (Haefliger, von Krogh, & Spaeth 2007; Messerschmitt & Szyperski 2003: 342-343). Software reuse is important for improving software quality and productivity (Mili, Mili, & Mili 1995: 528). Expected benefits of source code reuse are lower development costs, higher productivity, reduced cycle time, lower training costs, easier maintenance, higher quality, lower risk, and better interoperability (Poulin 1997: 7). The most important reason for reuse in the NASA study was saving time, followed by ensuring reliability, saving money, and lack of experience (NASA Earth Science Data System Reuse Working Group 2006). Based on a literature review Sametinger (1997: 11-14) argues that code reuse increases quality (error fixes accumulate from reuse to reuse), increases productivity (less code has to be developed), increases performance (reuse makes it worthwhile to invest more resources), increases reliability (reused software is well-tested), increases interoperability (same standards and interfaces), increases rapid development, and increases knowledge sharing (by reading the source code of other developers). At the same time code reuse decreases redundant work, time to market, documentation costs, maintenance costs, training costs, and team size. In fact, a study report for two reuse programs at Hewlett-Packard a reduction of defect density in the source code of 24% respectively 76% and an increase in productivity of 57% respectively 40% (Lim 1994: 24-25).[256] The time to market was reduced by 42% in one of the programs while for the other program no estimate is available.

In the NASA study the greatest barrier out of 10 to reuse source code was that the source code was difficult to understand respectively poorly documented (NASA Earth Science Data System Reuse Working Group 2006; see also Raymond 2004: 385). Raymond (2004: 381) observes that another barrier for code reuse is that programmers want to be artists. Because artists need a focused audience programmers do not want to reuse code but that their code is reused. According to Sametinger (1997: 15-18) there are i) managerial and organizational obstacles (e.g. legal issues, not invented here syndrome, lack of management incentives), ii) economic obstacles (costs of making source code reusable, costs of reusing source code, and costs of installing

[256] For other examples see Sametinger (1997: 14-15).

reuse processes), and iii) technical obstacles (e.g. difficulty of finding reusable software, integration causes problems, modification causes problems). Lynex and Layzell (1998: 106) report that a big problem is knowing how to obtain a component repository. They report also disincentives of code reuse for developers and management similar to Sametinger (Lynex & Layzell 1998: 109-111). There are also examples of major software failures based on software reuse such as for example the Therac-25 (Leveson & Turner 1993: 20) or the Ariane-5 incident (Nuseibeh 1997). Based on the Ariane-5 disaster Nuseibeh (1997: 15) concludes that source code “reuse does not necessarily increase safety, as the reuse of the ‘wrong’ piece demonstrated.” Therefore, it can be suggested that code reuse is basically beneficial but not trivial.

The permission to run, study, modify, and distribute the source code provides a very good condition for the reuse of source code across organizations. In consequence, software under a FOSS license is perfectly suited for source code reuse. This clear legal situation for FOSS code reuse does not seem to apply for commercial software development (Sametinger 1997: 39). The rights and responsibilities of providers and consumers of reusable components are difficult to specify. The owner of the source code components respectively the owner of the intellectual property has to agree that it is reused by other actors.

In commercial software development many different legal actors may contribute to a project. If one actor intends to reuse his part this may be very troublesome because he has to isolate his source code from the one of the other actors. Such problems do not exist for FOSS.

In the FLOSSPOLS study it was asked what they have learned from their participation in FOSS development (Ghosh & Rüdiger 2006: 21). Interestingly, the technical skill learned the most was how to reuse code written by others. To write source code in a way so that it that can be reused by others was also an important skill learned. Writing source code so that it is understood by others is important because of public peer review. If others do not understand the source code they will not accept it.

It can be summarized that talking about source code reuse without talking about FOSS is hardly possible (Raymond 2004: 381). The legal problems related to software reuse do hardly exist for FOSS applications. Because everybody knows that the source code is reviewed by others programmers write it more carefully. Therefore, some of the problems of source code reuse are smaller in FOSS projects than in commercial projects.

4.3.9 Coordination mechanisms

Some software projects are beyond the ability of an individual developer to create which makes coordination necessary (Kraut & Streeter 1995: 69). Individuals produce more if they cooperate with each other, if they specialize, and if they transact with each other (Milgrom & Roberts 1992: 25). Yet back in the seventies the ground control software for the Apollo spacecraft had about 23 Million lines of code that required approximately 11’500 staff-years of effort (Kraut & Streeter 1995: 70).[257] The consequent division of labor leads to a coordination problem. Only after Friday arrived Crusoe had to deal with coordination (Milgrom & Roberts 1992: 26, for a description of a Crusoe economy see chapter 6). Coordination is “integrating or linking together different parts of an organization to accomplish a collective set of tasks” (Van de Ven & Koenig 1976:

[257] This is only a rough estimation showing the magnitude. Amor-Iglesias et al. (2005: 15) estimate that the Linux distribution Debian 3.1 has about 230 million lines of code representing 60’000 person-years. Microsoft Windows XP is thought to have about 40 million lines of code (Amor-Iglesias et al. 2005: 16).

322). In software development coordination means that different developers working on a project agree on a common definition of what they are building, share information, and mesh their activities (Kraut & Streeter 1995: 69). Different organizations achieve coordination by different ways and with different results (Milgrom & Roberts 1992: 25).

Formal or informal coordination mechanisms?

In commercial software development a wide range of mechanisms such as managing requirements, design specifications, planning, measuring of process characteristics, regular status meetings to track progress, and implementations of workflow systems are used for coordination. Herbsleb and Grinter (1999: 2) argue that these mechanisms provide a shared understanding of what the purpose, the process, and the desired outcomes are. This should give the members of a team a common direction.

In FOSS development there are almost no formal coordination mechanisms such as regular status meetings or design specifications. Projects are mostly coordinated by some sort of muddling through. Coordination is therefore managed informally. For example peer review on contributed source code is implicitly a planning instrument (see chapter 4.3.1). Peer review reveals not only defects of the source code in question but also ideas about further development. A developer participating in the peer review may claim that a function X should be integrated in the future and that therefore Y has to be implemented in the reviewed source code. In many cases manpower is not assigned to tasks by a centralized entity but developers decide themselves on which task they work on (see chapter 4.3.2).

Goetz (2003) states that FOSS development "works like an ant colony, where the collective intelligences of the network supersede any single contributor". It parcels the work of a huge number of developers by using the distributive power of the Internet. Valverde et al. (2006) compare with an empirical social network analysis[258/259] self-organizing processes and the produced social hierarchies in FOSS communities on the one hand and in colonies of the European paper wasp on the other hand.[260/261] In both communities complex designs emerge from distributed collective processes. The involved agents have limited knowledge of the global pattern and decisions become localized and constrained by other project developments. The authors conclude that the two communities share some organization patterns.

Comparing coordination processes of commercial software projects and of FOSS projects two questions emerge: *Firstly*, does the self-organization approach in FOSS projects really work? By referring to the success of some FOSS projects one can answer this question definitely with yes (see chapter 2.10). *Secondly*, why are

[258] The authors analyzed for example centrality and link weight (Valverde et al. 2006).

[259] For a discussion on the relationship between collective action and social networks see Marwell and Oliver (1993: 101-129).

[260] The European paper wasp is a primitively eusocial species that cooperates in the care of the young, that has a reproductive division of labor, and in which there is an overlapping of different generations (Theraulaz, Bonabeau, & Deneubourg 1995: 314; Valverde et al. 2006: 37). The wasps live in small colonies of about 20 individuals. Their behavior is flexible and individuals tend to adopt specialized roles determined by *social interactions*. There is *no predetermined control* of individual activities. The hierarchy is stable, linear, transitive, and to some degree based on the wasp's age. Older wasps tend to be more dominant.

[261] For self-organization in ecosystems see also Camazine (2001), Hemelrijk (2005), and Solé and Bascompte (2006). For a distinction between individual tasks, group tasks, partitioned tasks, and team tasks in insect societies see Anderson and Franks (2001: 536) and for division of labor in such societies see Franks et al. (2001).

there no formal coordination processes in FOSS projects? In the following, this second question is discussed further.

Empirical studies (Curtis, Krasner, & Iscoe 1988: 1278-1282; Herbsleb & Grinter 1999: 2; Perry, Staudenmayer, & Votta 1994: 42-43) suggest that developers often use informal communication channels such as peer discussion; sometimes without management authorization or management knowledge. According to Curtis et al. (1988: 1280) programmers consider often communication in source code review processes as more important than its official purpose of finding defects. Curtis et al. (1988: 1269) report further that different studies demonstrate that special tools and development methods had a small and human and organizational factors a substantial impact on software productivity. Another study of commercial large scale software projects comes to a similar conclusion (Kraut & Streeter 1995). It showed that the most valued as well as the most used coordination mechanism is discussion with peers (Kraut & Streeter 1995: 74). The least used and the least valued mechanism are management case tools. At the same time subjects stated that the most valued and the most used source for help are other project members (Kraut & Streeter 1995: 76). Kraut and Streeter (1995: 80) find that the use of formal and impersonal modes of coordination such as project documents, bulletins, milestones, and delivery schedule was not associated with greater coordination success if controlled for project size. They conclude that this result suggests that formal procedures and processes are limited in boosting success.

Interestingly, although formal coordination mechanisms had little impact on coordination success they were valued highly by senior managers. Kraut and Streeter (1995: 80) suggest that senior managers are misled by an illusion of control. An example showing that the senior management does sometimes not fully understand its developers is the story of a now defunct major Linux distributor that laid-off a member of the developer team. The senior management thought that this developer was not very valuable because he burned only releases on CD's. It was thought that the task could also be done by a less-paid administrator without major programming skills The senior management but not the developers was puzzled that software releasing collapsed completely for weeks after the developer left the company. What happened? Burning CD's itself was not a very complicated task, indeed. However, doing so included merging the source code and using different complicated scripts. This part of the burning process was not trivial at all. The senior management could only see the observable process of burning CD's but not the complicated part.

It can be concluded that informal communication and self-organization play a critical role in coordinating software development in general (Grinter, Herbsleb, & Perry 1999: 307). In FOSS projects such informal communication and coordination methods are even more important. The lack of formal coordination processes in FOSS projects can be simply explained by two reasons: *Firstly*, formal coordination does not contribute much to the progress of development and it can be left out without drawbacks for performance. *Secondly*, in FOSS projects leaders are developers as well. Developers seem to have a strong preference for informal communication and coordination. In consequence, not much formal coordination is implemented in FOSS projects.

Making coordination obsolete

A proper way for dealing with coordination is making it obsolete. One way to do so is modularization (see chapter 4.3.6). Herbsleb (1999: 10) argue that in projects with a good design decisions about each component can be made in isolation from decisions about other components. Coordination about technical issues can therefore *partly* be made obsolete by good design.

Another mechanism to make coordination *partly* obsolete is the choice of a convenient license. Gambardella and Hall (2006: 880-882) argue that the GNU GPL license is an important coordination mechanism of FOSS development (see also licenses as constitutions in chapter 2.6). It removes ambiguity about the intentions of the other contributors. The GNU GPL urges contributors to reveal their modifications to the community irrespective whether they appreciate it or not. Additionally, a developer can be sure that his contributions will not be privately appropriated by others. License terms do not obsolete the need for coordination about technical issues. However, they obsolete *partly* the coordination of behavior of developers. The license terms govern what a developer is allowed to do and what he is not allowed.

Conway's Law

In an influential article on coordination in software development Conway (1968) writes that "any organization that designs a system … will inevitably produce a design whose structure is a copy of the organization's communication structure".[262] Brooks (1995: 111) described this phenomenon as "Conway's Law". Conway's (1968) example is an organization with eight people who have to produce a COBOL and an ALGOL compiler. Because of estimates of difficulty and time five developers are assigned to the COBOL job and three to the ALGOL job. As a result of this project structure the COBOL compiler runs in five phases while the ALGOL compiler runs in three phases. Following Conway's argumentation it could then be stated that source code structure follows project structure. Applied to FOSS projects one could guess that the degree of modularization depends not only on technological imperatives but also on the composition of the developer community.

De Souza et al. (2005: 205) observe that the software structure influences development processes. FOSS project structure follows then the source code structure. Parnas (1972: 1052) discusses how "modularization as a mechanism for improving the flexibility and comprehensibility" influences the characteristics of an organization and development processes. Modularization provides boundaries and is in consequence a mean for the division of labor. Modules are designed so that coordination is minimal and focused on interfaces. It seems to be reasonable that these technical structures are reproduced in the organizational structure. If a product has two modules A and B and not much coordination is needed after interfaces are designed it seems to be naturally that one team is working on module A while another teams is working separately on module B. Sanchez and Mahoney (1996: 64) state consequently that *modularity in product design* may influence *modularity in organization design*. It can be summed up that source code structure influences development processes respectively structures and vice versa.

4.4 FOSS principles beyond software development

Meyer (2005: xv) concludes that FOSS development is the "richest example" of the transformation from an industrial to an informational economy. While financial capital replaced land as the main store of economic

[262] There is a big amount of literature on how respectively whether technology influences organizational structures (for an overview see Gerwin 1981). While the Aston group (Hickson, Pugh, & Pheysey 1969) claims that technology is less important for determining the structure than other variables such as organization size Aldrich (1972) concluded that technology is an important variable for explaining organizational structure.

value in the transition from an agrarian to an industrial society it is knowledge that replaces financial capital in the transition from an industrial to an information society.

The FOSS production model is not limited to software production (Fitzgerald & Feller 2001: 273). Rather it is a mode of development for knowledge workers in the information society in general. In an article on FOSS development Goetz (2003) argues that this production model "is doing for mass innovation what the assembly did for mass production". He argues that this production model will spread from software development to other disciplines and from hard science to liberal arts. According to Meyer (2005: xviii) FOSS development is an archetype of a future production model because of two properties: *Firstly*, its self-organization mobilizes human capital. *Secondly*, the costs of reusing and distributing the products are low. Faber (2002) argues that not only product development but also education should adopt some characteristics of FOSS development such as user testing, releasing early and often, and developing in collaboration with users. In the following, a few examples of FOSS structures and mechanisms beyond software development are presented.

Web 2.0

The use of FOSS development principles for web-based services is described by the buzzword Web 2.0. A characteristic of Web 2.0 is that users own and control the data (Hinchcliffe 2006). It is based on an architecture of participation and democracy, social networks, and collective intelligence respectively wisdom of the crowd (Graham 2005; O'Reilly 2005). A technology that allows such online collaboration is the wiki principle. The wiki principle allows every user (in most cases also unregistered) to add, remove, or change content of a website.[263]

The Wall Street Journal as well as Business Week argue that the wiki principle is an interesting tool for commercial firms (Hof 2004: 128; Swisher 2004: B1). Examples of commercial firms already using wikis are Walt Disney, SAP, and Motorola (Hof 2004: 128). Indeed, a study on corporate wiki users indicate that very different companies from less than 100 employees to more than 10'000 employees use wikis (Majchrzak, Wagner, & Yates 2006: 100, see also chapter 3.2). A subject who participated in the study illustrated the use of wikis in his or her company in the following way (Majchrzak, Wagner, & Yates 2006: 100-101):

> *The Sales department uses wikis to log the daily lead counts and to get information about partnerships, positioning, product features, and company intelligence; the Professional Services department uses the wiki to outline the details of each client implementation and record progress as miniprojects; the Operations department uses the wiki to update the company on product issues; the Marketing department uses the wiki to produce web and print collateral and manage some aspects of marketing campaigns; the Product Management department uses the wiki to track interesting marketing trends, and the Partnerships department uses the wiki to collaborate on a joint project with a remote team at another company in a different continent.*

Wikis are thought to increase knowledge reuse, increase collaboration efficiency, and improve work processes (Majchrzak, Wagner, & Yates 2006: 101). At the same time they are not thought to be useful to identify new

[263] Wikis are obviously very vulnerable to vandalism. In fact, the Wikipedia project recognizes 25 different kinds of vandalism (Wikipedia 2007d). Astonishingly, a study found that for some sorts of vandalism such as mass deletions the median time until the modifications were fixed is below 3 minutes (Viégas, Wattenberg, & Dave 2004: 579). Although mass deletion is very easy to detect in contrast to other types of vandalism such as "sneaky vandalism" (adding misinformation, changing dates) the related reaction time is notwithstanding notable.

business opportunities. Interestingly, some types of motivation to contribute to corporate wikis such as peer recognition, reputation, and personal requirements are similar to FOSS development.

The most successful and prominent example of a wiki is the online encyclopedia Wikipedia. Depending on the study Wikipedia ranks currently between the ninth[264] and eleventh[265] most-visited website worldwide. It has roughly 5 million articles in 229 languages (Wikipedia 2006g).[266] Wikipedia is based on the wiki principle in which everybody is able to add, remove, or edit content by using a web browser.

In Wikipedia there are different contributors such as non-registered contributors, registered contributors, developers, stewards, bureaucrats, and administrators (Wikipedia 2006f).[267] In June 2006 there were 214'360 contributors who made at least 10 edits since they have registered (Wikipedia 2006g). Out of these Wikipedians 67'148 contributed more than 5 edits in June 2006 and 9258 contributed more than 100 monthly edits.[268] Based on this statistics it is clear that active (more than 5 edits a month) and very active (more than 100 edits a month) contributed heavily to the 8 Millions edits in June 2006. The assumption that these active and very active contributors keep Wikipedia running is only partly correct. In Wikipedia usually a non-registered outsider makes one edit to add information (Swatz 2006). An insider respectively a registered contributor makes then several edits tweaking and reformatting it. These insiders cumulate thousands of edits by doing things like changing the name of a category. While insiders account for the vast majority of the edits outsiders provide much of the content. It can be concluded that a mix of different types of contributors keeps Wikipedia running.

In Wikipedia decisions on content and editorial policies are largely made through consensus decision making and sometimes by vote (Wikipedia 2006f). The founder Jimmy Wales retains a final judgment on policies and user guidelines. However, each language has a different culture regarding the power structure (Wikipedia

[264] According to the Netcraft list Wikipedia is after Google, Yahoo, Microsoft, BBC, FOX News, eBay, CNN, and Netcraft the *ninth* most visited site on the World Wide Web [Netcraft, 2007 #4881. However, Wikipedia is not the ninth most visited URL because some sites, especially Google, have several URL's such as for example http://www.google.com and http://www.google.de. Such URL's are only counted as one site that is technically not correct but is good enough for the presented argument. At the same time the data for Wikipedia includes only the English site. The data is based on the sites accessed through the free anti-phishing Netcraft toolbar for Firefox and Internet Explorer browser. Although this tool is common it clearly introduces a selection bias. This is apparent in the result for the Netcraft website that is the eight most visited website. Nevertheless, the basic argument that Wikipedia is a widely frequented website remains.

[265] According to Alexa [, 2007 #4882], a subsidiary company of Amazon.com, Wikipedia is the eleventh most visited site. The sites in front of Wikipedia include Yahoo, Microsoft, Google, YouTube, Myspace, Baidu, Orkut, and QQ. The data is based on the sites accessed (average over the last three months) through the Alexa Toolbar integrated in Firefox and Internet Explorer browsers. Therefore the sample bias is similar to the one discussed for the Netcraft analysis of web traffic (see footnote 264). The top position of Baidu (a Chinese web search engine) is for example partly due to the popularity of the Alexa Toolbar in China (for a short overview over methodological problems see Alexa 2007).

[266] The five languages with the most articles (English, German, French, Polish, and Japanese) account for more than half of all articles (Wikipedia 2006h). Languages such as Wolof, Tetum, Udmurt, or Voro have only a few hundred articles.

[267] Developers write software or maintain the Wikipedia servers. Administrators or sysops maintain the Wikipedia content by protecting article, deleting articles, or blocking IP addresses. Bureaucrats are Wikipedia users who are able to promote other users to administrators or bureaucrats, who grant or revoke a user's bot status and may rename a user account. Stewards are similar to bureaucrats but may also promote users to stewards.

[268] The most active contributor (without bots in main namespace, edits in all namespace in parenthesis) made 16'353 (17'124) edits in March 2006 (own calculation based on Wikipedia 2006i). On average the ten most active contributors made 7823 (9620), the 100 most active contributors 2703 (3865), and the 1000 most active contributors 768 (1232) edits in March 2006. The 1000 most active contributors made together 768'268 edits of total 6.9 millions edits in March 2006 (Wikipedia 2006g). This is more than 11% of all edits.

2006d).[269] The power structure is a mix of anarchic, despotic, democratic, republican, meritocratic, plutocratic, and technocratic elements (Wikipedia 2006d). However, Kittur et al. (see also Ball 2007; 2007) argue that Wikipedia has mutated from an oligarchy to a democracy. From late 2002 to 2006 the percentage of all edits made by administrators (= elite users) decreased from 59% to 10% (Kittur et al. 2007: 2).

The power of user involvement can be impressively shown by the history of Wikipedia. Being an admirer of Linus Torvalds and Richard Stallman Wikipedia founder Jimmy Wales had the vision to use volunteer contributions to create an Internet encyclopedia (Goetz 2003). His first attempt was in 1999 when he started the project Nupedia. Contributors had to apply for access, articles were peer reviewed, and professionally edited. The editorial process consisted of seven different steps and editors were expected to possess a Ph.D. (Wikipedia 2007b). *After two years* the project had *12* articles. In 2001 Wales started the project Wikipedia and eliminated the barriers of participation by using the wiki principle. *After two years* Wikipedia had about *20'000* articles, after two further years about 200'000, and after another two years almost 1'000'000 articles (Wikipedia 2006j).

Project Gutenberg

FOSS development principles are not only exerted in wikis or Wikipedia but also in other projects. For example in the Project Gutenberg volunteers digitize, file, and distribute books for which the copyright has expired (Project Gutenberg 2006). The project was founded in 1971 and in August 2006 over 19'000 books were available. Each day approximately 400 volunteers do some work in the project such as proofreading scanned books.

NASA Clickworkers

NASA Clickworkers was an experiment in which users could mark Mars craters online without being paid (Benkler 2002: 384). The mapping task is incremental and takes in the minimum only a few minutes. The average mapping result of 200 Clickworkers was compared to the result of a trained geologist with many years of experience in identifying Mars craters. Astonishingly, there was no distinguishable difference between the inputs of the Clickworkers and the geologist. In the first six month of operation 85'000 users contributed 1.9 million entries. Interestingly, more than one third of the work was done by onetime contributors.

Galaxy Zoo

Yet another example of citizen science is Galaxy Zoo, an online astronomy project in that volunteers classify galaxies. The project is a tremendous success: Within less than a month 85'000 participants have submitted over 12 million galaxy classifications (Galaxy Zoo 2007). There is no rewarding apart from being the very first human being who sees a certain galaxy. By classifying the galaxy a Dutch Galaxy Zoo user found Hanny's Voorwep, a very strange astronomical object that attracted an enormous attention from the scientific community (Nemiroff & Bonnell 2008).

[269] For an interesting paper on the motivation to contribute to the German Wikipedia and for a comparison to FOSS development see Shroer and Hertel (2007).

No online interaction but under a FOSS alike license

In the examples discussed above online interaction was an important aspect. Indeed, it seems to be plausible that the FOSS development model is especially suited for goods that are produced by online interaction. However, there are also some examples in which products or services under a FOSS license are not respectively not yet produced online. An example is the UniProt (Universal Protein Resource) database.[270] UniProt is a central repository of protein data under a FOSS alike Creative Commons license (Apweiler et al. 2004; Leinonen et al. 2004; Wu et al. 2006). It is according to a *self-assessment* “the world's most comprehensive catalog of information on proteins” (UniProt 2006). Giles (2007: 691) calls it in a Nature article a key database in biology.

Similarly, the results of the Human Genome Project that identified all human genes and the complete sequence of the 3 billion DNA subunits are in the public domain (Goetz 2003; Human Genome Project 2006; Nature 2001).

User innovation for physical goods

In an analysis of agricultural equipment Douthwaite et al. (2001: 853) found that insufficient involvement of users respectively key stakeholders may lead to technological failure. Von Hippel (2005b: 267-268) argues that user innovation does not only work for information goods such as software but also for physical goods (see also 0). An example is high-performance windsurfing (von Hippel 2005a: 268-269). For von Hippel (2005a: 93-95) user innovation is a widely distributed process. In this process the users have a set of innovation related needs that makes it beneficial for them to contribute. Innovation networks of users flourish only if i) some users have a sufficient incentive to innovate, ii) some users have an incentive to reveal their innovation, and iii) the diffusion of the innovation by users can compete with commercial production and diffusion (von Hippel 2005b: 270).

In two subsequent studies von Hippel (1988: 11-27) studied the sources of innovation. He found that for scientific instruments 77% of all innovations were developed by users. For all of the four researched scientific instruments the initial innovation was made by users. About 82% of major and 70% of minor improvements related to the scientific instruments were made by users. Because users of scientific instruments are probably not typical users von Hippel also studied the manufacture of silicon-based semiconductors and the assembly of printed circuit boards. He found that in this area 67% of all innovations were made by users. Again, all initial innovations were made by users while they made 63% of major and 59% of minor improvements.

[270] However, the UniProt database has been connected to a wiki to foster a collaborative approach and to keep up with discoveries (Giles 2007: 691).

Percent of innovation developed by users					
		All innovation	Initial innovation	Major improvement	Minor improvement
Scientific instruments	Gas chromatograph	82%	-	-	-
	Nuclear magnetic resonance spectrometer	79%	-	-	-
	Ultraviolet spectrophotometer	100%	-	-	-
	Transmission electron microscope	79%	-	-	-
	All scientific instruments	*77%*	*100%*	*82%*	*70%*
Process Machnineries	Manufacture of semiconductor	71%	100%	71%	56%
	Assembly of printed circuit board	63%	100%	40%	63%
	All processes	*67%*	*100%*	*63%*	*59%*

Figure 49: User innovation[271]

User innovation for physical goods: The LEO example

The world's first business computer LEO was manufactured by the British catering and food-manufacturing company J. Lyons and Company (Aris 2000; Caminer 2003; Hendry 1987; Land 2000). That a food company as a user of electronic equipment and not an electronic firm was the leader in this high technology field is quite interesting and an astonishing example of user innovation.

LEO was used to administer payrolls, invoicing, stock control, and production scheduling (Aris 2000: 4). A problem of building LEO was that business computing was not similar to scientific computing or militaristic computing that were characterized by little data, much computation, and few results (Caminer 2003: 6). For business purposes the computer had to handle much input and much output. The challenge was rapid feed in of data and the recording of the results. In fact, this resulted in the original intention to use magnetic tapes and not punch cards which was different to computing for science or the military (Aris 2000: 6). Having a backup was also more important in business computation than in other areas such as for example ballistic table computation for which ENIAC was built. Therefore, the J. Lyons and Company itself as a user had to develop much of the hardware and all software from scratch.

Historical antecedents

Osterloh and Rota (2007) argue that the FOSS development model has historical antecedents. They show many parallels to what Allen (1983) calls collective innovation (see also Henkel 2006: 3). However, Osterloh and Rota (2007: 169) argue that there are two main major differences. *Firstly*, collective invention hardly survives the development of a dominant design. At this point commercial applications get profitable. In consequence, individuals started to keep their innovation private to appropriate profits that in turn terminate collective innovation. The FOSS development model solves this problem by the copyleft mechanism that

[271] Von Hippel (1988: 15 and 22).

prevents that derivative work is taken private. *Secondly*, the characteristics of FOSS development create a low-cost situation for contributing.

4.5 Excursion: The power of self governance or pirate democracy vs. merchant ship tyranny

A striking example how governance mechanisms may heavily influence organizational outcome are pirate communities in the late seventeenth and early eighteenth centuries.[272] It shows how hierarchical instruction in contrast to self-governance may hinder performance. It is also a didactic play for showing that procedural fairness as a prerequisite for successful cooperation is even important for harsh individuals.

In popular stories pirates are portrayed as drunken, cruel, and rough sailors what they in fact were. However, another aspect of pirate life is hardly known. The pirate community around the free colony Libertalia at Madagascar was during the time of absolute monarchs one of the few true democracies (Sherry 1986: 122).[273] Countries like France or Russia were ruled at that time by the absolute monarch Louis XIV, known as The Sun King, and the powerful Peter I, known as Peter the Great.[274] These hierarchically organized structures in France or Russia were reproduced on merchant ships. The subsequent tyranny of the captains on merchant ships was, however, in sharp difference to the democracy on pirate ships (Cordingly 1997: 96) which will be briefly discussed below.

On pirate ships democratic principles were assured by the ship articles. They were in fact the constitution of the ship that spelled out the rules of behavior, duties, and the powers. The ship articles were initiated by majority vote at the start of a big journey. Although the intentions of the different article were mostly similar, surviving ship articles differ in detail (Rogozinski 2000: 172). They were not copies of old articles from prior voyages. In fact, they were always worked out anew before the journey started. Some articles resembled miniature constitutions (Rogozinski 2000: 171). A typical example of a ship article is the one from the Bartholomew Roberts's ship that starts with the following object (Sherry 1986: 124-125):

> *I. Every man shall have an equal vote in affairs of moment. He shall have an equal title to the fresh provisions or strong liquors at any time seized, and shall use them at pleasure unless a scarcity may make it necessary for the common good that a retrenchment may be voted.*

The crew of a pirate ship believed to be self-employed and collective owners of their ship (Sherry 1986: 122). It was therefore taken for granted that everybody had an equal right to voice his opinion and an equal vote in decisions.[275] Almost everything was subject to collective decision making, for example whether to fight or not or where and when to anchor. Every pirate had the right to object every decision. Only during battles this right and democratic procedures were abandoned (Sherry 1986: 128). Engaged in fighting, the captain turned into the absolute master. He was allowed shooting anyone who denied his command. The judicial system rested at any other time on public trials and majority decisions (Sherry 1986: 125). However, as deviation from this rule the quartermaster had the right to order punishment for *minor* offences. His position was almost

[272] See footnote 5.

[273] The only other places in Europe with a democratic rule similar to Libertalia were the Swiss cantons (Rogozinski 2000: 261).

[274] Formally, after 1689 Peter I co-reigned with his half-brother Ivan V. However, it seems that Ivan V was powerless.

[275] See the section on fairness in chapter 5.4.2.1.2.

as important as the one of the captain and served as a counterbalance to him. The quartermaster had the important task to decide which plunder was to be taken from the prize and to distribute it fairly.

Based on merit captains, quartermasters, and other officers were elected and deposed by the crew with majority (Rogozinski 2000: 139). Defoe (1724) reports a pirate community that elected thirteen different captains during a few months. Although firing a captain was always possible this step was usually not taken lightly (Rogozinski 2000: 176). The pirate captains and officers wore no special uniforms and had no major privileges (Sherry 1986: 123). Even though captains had sometimes a cabin of their own as a mark of their crew's esteem, any crewman could enter anytime and make use of all furnishings. Rogozinski (2000: 173-174) argues that pirate captains had in contrast to the ones at naval or merchant ships often a very strong, assertive, and commanding personality. To maintain authority and order they could not rely on whipping the crew as done at merchant and navy ships. In fact, pirate captains had to motivate pirates to join the cruise voluntarily and to convince them that his suggestions are appropriate. Pirate captains could also not rely on privileges based on their social heritance.

The plunder of the pirates was shared almost equally (Sherry 1986: 122). Captains, quartermasters, and officers got only slightly more as it is stated in the Bartholomew Roberts's ship article (Sherry 1986: 125):

> *X. The captain and the quartermaster shall each receive two shares of a prize … all other officers one and one quarter and private gentlemen of fortune one share each.*

This distribution key was accompanied by something similar to an insurance system. The compensation for injuries was determined by different parts of the body (Cordingly 1997: 97). In one of the ship articles 600 pieces of eight was awarded for the loss of the right arm, 500 pieces of eight for the loss of the left arm, 100 pieces of eight for the loss of an finger et cetera (Cordingly 1997: 97).

Life on merchant ships was quite different from life on pirate ships. Captains possessed in fact the power over life and death (Sherry 1986: 53). Mistreatment of sailors was an integral part of life on these ships. It was far from unique to flog a man if he was bold enough to look at the captain. Whipping a sailor to death for loosing an oar was also nothing special. In the Pacific on the average more than one fifth of the crew was flogged during a journey (Dening 1992: 114).[276] After watching the famous film about the mutiny on the Bounty one may think that the captain William Bligh was especially violent and that this caused the revolt. On the contrary, he was milder in displaying physical force than any other merchant captain at that time (Dening 1992: 62). He flogged on his two voyages to the Pacific fewer sailors than every other captain in the Pacific did in the whole eighteenth century. Physical punishment seen in films about the Bounty is not an expression of special cruelty of the captain but of the regular means of command. Captains at that time "had absolute power over the officers and seamen of their ships, they could make life as tolerable or unbearable as they wished" (Rogozinski 2000: 166). Captain tyranny was backed by law and upheld by the Admiralty Courts. Additionally, food and accommodations at merchant ships were horrible (Sherry 1986: 53-54). Life was also dangerous. In wartime merchant ships were expected to capture, like pirates, enemy ships. However, in contrast to pirates they did not get a fair share of the prize. Life for a merchant sailor was also unsafe in peacetime because of fights with pirates. The pirates expected and demanded instant surrender from merchant ships. If the merchant ship sailors did so, the pirates "usually refrained from inflicting violence on the crew" and they were in this case "almost gentlemanly in their behavior" (Cordingly 1997: 120). It is therefore not

[276] Apart from beatings with the rattan rope physical punishment had to be officially recorded (Dening 1992: 117).

astonishingly that the great majority of merchant ships surrendered immediately without fighting if attacked by pirates (Cordingly 1997: 121). There was no motivation for the sailors to defend the property of people who despised and abused them. It seems to be quite obvious that they felt no obligation to give their lives for a cause that was not theirs. In contrast, the pirates had the unusual right of personal liberty[277] and got a fair share of the prize.

Because at that time the class system was inflexible a sailor could not expect to ever change his miserable life. There was one exception: one could get a pirate with the result of having individual rights and the possibility of getting rich. Pirates “were thoroughly egalitarian and refused to pay the customary respect to social rank and status” (Rogozinski 2000: 140).[278] The success of the pirates seems to be partly based on self-regulation and personal liberty that were in sharp contrast to merchant ship tyranny and the strong hierarchy everywhere else.

Interestingly, the governance structures and mechanisms of pirate communities and FOSS communities show a few correspondences. Meritocracy is in both communities very important. Attracting followers does in both communities not rest on hierarchical authority but on finding acceptance. Community members in FOSS projects can leave their project whenever they want. The same applies similarly to pirates who could leave their ship (almost) whenever they wanted. The importance of fair processes such as voting is another match. In both cases the community members are respectively were also highly motivated to struggle for their cause. This comparison leads to a very important implication. If such different communities are able to adopt similar structures and mechanisms they must be to some extent universal. This in turn suggests that other organization also may adopt some of the FOSS mechanisms and structures.

4.6 Summing up

The basic insights of this chapter are:

- Due to division of labor FOSS projects need coordination as well as governance structures and mechanisms. > See chapter 4
- Next to luck factors responsible for the success of Linux were the GNU GPL license, the incorporation of other programmer’s input, and early and often releases. Factors for the problems of the NetBSD project were weak leadership, ownership of source code, poor public code reviews, the behavior of a commercial firm, and inefficient executive committees. > See chapter 4.2
- Peer review and egoless programming are important mechanisms of FOSS development. They lead to more understandable source code, to concrete feedback for developers, to project information for outsiders, and to project documentation. They also serve as a basis for decision making for project leaders and as a communication tool respectively as a planning instrument. > See chapter 4.3.1
- Developers choose themselves to which project they want to contribute. Because of this self-selection they contribute to tasks that are appropriate to their skill level. This is important because software

[277] For black pirates this included liberation from slavery.

[278] In fact, Dening (1992: 154) argues that the violence of the officers to their crews had little to do with running the ships efficiently but with class and status.

development depends on creativity, because there is high variability among individuals, and because resources are difficult to assess from the outside. > See chapter 4.3.2

- In FOSS development interpersonal interaction is computer-mediated and collaboration is organized online. There is no or little face-to-face communication. However, this does not seem to have important consequences. > See chapter 4.3.3
- To *release early* is an important aspect of FOSS development because prospective developers need a plausible promise that there is a chance for project success. To *release often* respectively incremental development has three advantages. *Firstly*, it motivates developers because they see fast that their efforts pay off. *Secondly*, incremental development masters complexity. *Thirdly*, small and therefore cheap contributions are possible that increases the likelihood of privileged users. > See chapter 4.3.4
- The inclusion of users as developers is very basic to FOSS development. It is important because some bugs are hard to find if they are searched for intentionally. > See chapter 4.3.5
- Modularization as a principle of FOSS development facilitates code reuse, makes the use of different programming languages possible, reduces complexity, and prevents strange interactions. It also allows parallel development, efficient system upgrading, and to break the project into bite-sized tasks. However, there are also limits to modularization. Designing interfaces and integrating modules are not trivial and interdependencies may emerge which were not considered at first. > See chapter 4.3.6
- Decision making in FOSS development is either made by a benevolent dictator or based on democratic rules. There are no indices which type of decision making is more adequate. However, in either case the importance of proved merit and talent of leaders is high. > See chapter 4.3.7
- Code reuse as practiced in FOSS projects makes development very efficient. > See chapter 4.3.8
- FOSS projects are coordinated informally and self-organization plays a critical role. The lack of formal coordination does not seem to have disadvantages. The analysis of commercial software development shows that developers perceive formal coordination as not very useful. A proper way of dealing with coordination problems is making them obsolete. Possibilities to do so are modularization and choosing a convenient license. > See chapter 4.3.9
- The characteristics of source code may influence the structure of a project and vice versa. > See chapter 4.3.9
- The FOSS production model is not limited to software. Rather it may be a mode of development for knowledge workers in general. Related examples are corporate wikis, Wikipedia, the Project Gutenberg, the NASA Clickworkers, the UniProt database, the Human Genome Project, the first business computer LEO, scientific instruments, silicon-based semiconductors, and the assembly of printed circuit boards. That some of the FOSS development principles are not limited to software is supported by the example of self-governed pirate democracies. > See chapter 4.4 and 4.5

This chapter has discussed FOSS governance structures and mechanisms. For the ongoing argumentation there are three main insights. *Firstly*, FOSS governance structures and mechanisms are multifaceted. *Secondly*, the appropriate selection of structures and mechanisms are crucial as the examples of Linux and NetBSD show. This urges to include corresponding hypotheses in this dissertation. *Thirdly*, the discussed governance structures and mechanism are not limited to FOSS and software development. This suggests in consequence that the drawn lessons may be partly valid beyond software development.

5 Social dilemmas: The basic challenge

5.1 Where social dilemmas come from

In the following subchapter it is argued that a society cannot completely rely on private goods and self-interest of individuals. It is briefly set forth that it is inevitable for individuals of a society to share common goods. However, in consequence social dilemmas arise. This dilemma will be briefly discussed and exemplified for the prisoner's dilemma.

Private goods

In his seminal book "An inquiry into the nature and causes of the wealth of nations" Adam Smith (1976: book 1, chapter 2, 18) stated that it "is not from the benevolence of the butcher, the brewer, or the baker that we expect our dinner, but from their regard to their own interest". Following this heavily cited quotation it is often recommended that nations and organizations should rely on individual rationality to maximize prosperity. However, while citing Smith and favoring rational egoism it is often neglected that in his example private goods are traded. People not willing or able to pay can be excluded from eating sausages and bread or drinking beer. Although many goods do feature this attribute clearly other goods do not. Sometimes individuals cannot be excluded from benefiting from goods. Regarding his own interest, the baker would not produce loaves of bread if people who do not pay could not be excluded. In this case, prosperity is not maximized and the invisible hand disappears (Buchanan 1967: 115). From an economic perspective this results in equilibrium outcomes that are Pareto inefficient (Kopelman, Weber, & Messick 2002: 113). Therefore, reliance only on self-interest of the baker is not always the best way providing prosperity. Selfish behavior is also not the only psychological motive of man (Smith 2000: 3, part 1, section 1, chapter 1):

> *How selfish soever man may be supposed, there are evidently some principles in his nature, which interest him in the fortunes of others, and render their happiness necessary to him, though he derives nothing from it, except the pleasure of seeing it". Therefore he concludes that utility is only one "of the principal sources of beauty".*

State of nature versus having something in common

The question of goods that are not private has been discussed for a long time. Already Aristotle (1912 Book 2, chapter 1) concluded:

> *To have nothing in common {for members of a state} is evidently impossible, for society itself is one species of; and the first thing necessary thereunto is a common place of habitation, namely the city, which must be one, and this every citizen must have a share in.*

If individuals have nothing in common Hobbes (1998: 84) predicts the famous "state of nature" in which no civilized society may emerge:

Whatsoever therefore is consequent to a time of war, where every man is enemy to every man, the same consequent to the time wherein men live without other security than what their own strength and their own invention shall furnish them withal. In such condition there is no place for industry, because the fruit thereof is uncertain: and consequently no culture of the earth; no navigation, nor use of the commodities that may be imported by sea; no commodious building; no instruments of moving and removing such things as require much force; no knowledge of the face of the earth; no account of time; no arts; no letters; no society; and which is worst of all, continual fear, and danger of violent death; and the life of man, solitary, poor, nasty, brutish, and short.

In the state of nature man is the absolute lord of his own person and possessions but this empire is in fact constantly exposed to the invasion of others. In consequence it may be reasonable for the individuals to give up their total freedom (Locke 1690, § 123).

Social dilemmas

Establishing a social contract to get over the state of nature and creating a good common to everybody is not without problems (Aristotle 1912 Book 2, chapter 3):

… *what is common to many is taken least care of; for all men regard more what is their own than what others share with them in, to which they pay less attention than is incumbent on every one* …

The problem described by Aristotle can be reformulated as a social dilemma that has two properties (Dawes 1980: 169):

i. Each individual receives a higher payoff for a socially defecting choice than for a socially cooperative choice. This holds true no matter what other individuals do.

ii. All individuals are better off if all cooperate than if all defect

This dilemma can be framed as a special version of an externality (Andreoni 1995b; Cornes & Sandler 1996: 4). Buchanan (1971: 7) noticed that "an externality as being present whenever the behavior of a person affects the situation of other persons without the explicit agreement of that person or persons". In social dilemmas an individual’s payoff depends on other’s decision (see also Schelling 1978: 213). Additionally, it is assumed that individuals are free to choice whatever action they prefer.

The prisoner’s dilemma

A classical adaptation of a social dilemma is the prisoner’s dilemma. Straffin (1980: 103) asserts that “any modern discussion of the meaning of rationality in social behavior must come to terms with the prisoner’s dilemma”. Below, the prisoner’s dilemma will be briefly discussed to present the social dilemma exemplarily.

The founder of the prisoner’s dilemma story describes it the following way (Tucker 1980: 101):[279] Two men (I and II) are charged with a joint violation of law. They are held separately and the police give both of them the following options:

[279] The prisoner’s dilemma was formally established by Merrill Flood and Melvin Dresher (1952; Flood 1958) at Rand. Albert Tucker (Tucker 1980) composed in 1950 the famous story to the prisoner’s dilemma that gave the game its name (Straffin 1980).

i. If one confesses and the other does not, the former will be given a reward of one unit and the latter will be fined two units.

ii. If both confess, each will be fined one unit.

At the same time, each of the two men has good reasons to believe that

iii. if neither confesses, both will go clear.

This situation gives rise to a simple symmetric two-person game with the following table of payoffs in which each ordered pair represents the payoffs to I and II in that order:

		II	
		confess	not confess
I	confess	(-1, -1)	(1, -2)
	not confess	(-2, 1)	(0, 0)

Figure 50: Payoffs in the prisoner's dilemma[280]

Apparently, for each man the pure strategy to confess dominates the strategy not to confess. If the other man confesses each man gets a higher payoff if they confess as well (-1 instead of -2). Similarly, if the other man does not confess it is better to confess (1 instead of 0). In consequence, both will confess and be fined with one unit. Hence, there is a unique equilibrium point given by the pure strategy to confess. However, if both men were to form a coalition and did not confess they would be clearly better off (both get a zero payoff).

The game becomes a zero-sum three-person game by introducing the State or society as a third player. The State cannot make a choice but receives payoffs as follows: The state gets a payoff of 2 if both confess. This is the best situation because both men can be punished without having to pay for rewards. If only one confesses the state benefits 1 unit (one men gets away unpunished) while if nobody confesses the state receives nothing.

		II	
		confess	not confess
I	confess	(2)	(1)
	not confess	(1)	(0)

Figure 51: Payoffs for the state in the prisoner's dilemma[281]

5.2 Wording

There is obviously some confusion about proper wording within research on social dilemma. Studies that focus on the produced good either research public or common goods. Scholars interested more in the decision

[280] Tucker (1980: 101).
[281] Tucker (1980: 101).

problem talk about social dilemmas. The ones mainly concerned about how to solve the problems research collective action. Still others are interested in the effects of underprovision or overuse of goods. While discussing the same phenomenon these researchers are interested in different aspects. Hauert et al. (2002: 1129) argue that "the diversity of the names underlines the ubiquity of the issue".

The different terms are specified here as follows:

- ***The decision problem:*** The basic decision problem faced by an *individual* is illustrated by the social dilemma.[282] Individuals can free ride or cooperate.
- ***The aggregate behavior:*** The *aggregate* behavior of individuals is entitled collective action.[283] Whether collective action succeeds or not depends on a nearly unmanageable amount of different factors such as individual differences, the decision structure, the social structure, and perception.
- ***The object:*** The points of reference of the decision in social dilemmas are *collective goods*.[284] If collective action is successful either *public* (Fulk et al. 1996) or *common goods* are produced. Attributes of these goods are non-excludability and for public goods additionally non-rivalry.
- ***The consequences:*** Depending on the aggregated behavior the collective goods are either *overused* respectively *underprovided* or they are used *sustainable respectively produced effectively*.

Decision problem	Aggregate behavior	Object	Consequences
Social dilemma	***Collective action***	***Collective good***	***State of use***
• Defecting / free riding • Cooperative behavior	• Individual differences • Decision structure • Social structure • Perception	• Public good (non-excludability and non-rivalry) • Common good (non-excludability)	• Underprovision or overuse • Collective effective and sustainable use

Figure 52: Terms overview

In the following, the presented wording will be used as structured above. However, as many other papers this study will not always succeed in keeping the wording consistent.

5.3 Definition and classification

Social dilemmas refer to the following situation (Dawes 1980: 169):

[282] For the relationship between individual respectively aggregate behavior and collective action see Marwell and Oliver (1993: 4).

[283] See footnote 282.

[284] Collective good as an umbrella term for common and public goods is not widely used (but see McDaniel & Sistrunk 1991). However, it is thought here that it is very suitable. For another usage of the term collective good see Olson (1995: 15).

i. Each individual receives a higher payoff for a socially defecting choice than for a socially cooperative choice no matter what other individuals do and

ii. all individuals are better off if all cooperate than if all defect.

Social dilemmas arise for two different forms of collective goods: public and common goods (Brewer & Kramer 1986: 543). In the case of public goods individuals must decide whether *to contribute* to a collective good. If the amount of contributions is too low the public good is not or only suboptimal produced. An example of a public good is the prisoner's dilemma that has already been described in chapter 5.1. Public goods are in general artificially produced goods (Stevenson 1991: 55).

If individuals have to deal with common goods they must decide whether *to take* from a collective good or not. The example best known is the tragedy of the commons described by Hardin (1968: 1244). A piece of grazing land is open to everybody (for an overview about common land see Zückert 2003).[285] A farmer has then to decide whether to put another cow on the commons. He may appropriate the benefits of the additionally produced goods completely for his own. However, the costs of overgrazing created by one more animal are shared by all farmers. The individual farmer has therefore only to bear a fraction of the costs he causes by putting an additional cow on the pasture. Because every farmer can privatize the benefits but collectivize the costs the pasture will be overgrazed and finally destroyed.

Although the decision problems related to public and common goods are framed differently, they are logically similar. Both goods rely on voluntary actions and individuals should refrain from selfish behavior.

Classification of collective goods

There are two relevant conditions for collective goods that are *non-rivalry of consumption* and *excludability* (Snidal 1979, see also Figure below). While only non-excludability is thought to be a necessary condition for common goods it is usually assumed that non-rivalry as well as non-excludability are necessary for public goods (Stevenson 1991: 54). However, there are arguments that similar to common goods non-rivalry is not a necessary condition for public goods.[286] Olson (1995: 14) explicitly states in his path breaking book "The logic of collective action" that jointness of supply respectively non-rivalry is not a necessary attribute of a public good. For others such as Samuelson jointness of supply is very important (Head 1977: 228).[287] In line with a majority of scholars it is assumed here that non-rivalry respectively jointness of supply is a necessary attribute of public goods.

For private goods there is rivalry in consumption as well as excludability. Excludability is thereby thought to be the more important characteristic because it allows owners to take advantage of their investments (Rose 1986: 711). There is no rivalry but excludability for club goods.

[285] Common grazing land was hardly open to *everybody*. Often only commoners were allowed to use common land (Stevenson 1991: 41,48, and 52). Ostrom (1997: 62) reports that in the Swiss communal alp Toerbel foreigners were not allowed to exercise any communal right. Therefore, one should not confuse open access with common property.

[286] In the following section the problems of non-rivalry will be discussed.

[287] For a discussion whether jointness of supply is a necessary attribute of public goods see Snidal (1979).

<table>
<tr><th colspan="2" rowspan="2">Classification of different goods</th><th colspan="2">Excludability</th></tr>
<tr><th>Yes</th><th>No</th></tr>
<tr><td rowspan="2">Rivalry</td><td>Yes</td><td>Private good
(shoes / homesteads)</td><td>Common pool resource
(common land / ocean fishes)</td></tr>
<tr><td>No</td><td>Club good
(toll bridges / pay TV)</td><td>Public good
(AM broadcast radio / firefighting)</td></tr>
</table>

Figure 53: Classification of goods

The difference between public and common goods is that for public good underprovision is the problem while overuse is the problem for common goods.[288] Social psychologists and ecologists have given more attention to common goods while sociologists and economists focused more on public goods (Brewer & Kramer 1986: 543). Because of the methodological diversity in these fields of study there are some differences in the research on public and common goods. Despite these differences individuals are faced with the same basic structure of the decision problem (Brewer & Kramer 1986: 543). In both situations individuals have less for themselves if they choose to act in the collective interest than if they had done otherwise.

In the following non-rivalry in consumption and non-excludability are discussed in detail.

Non-rivalry in consumption

Non-rivalry in consumption is also known as *jointness of supply* or *divisibility in benefits* (Snidal 1979: 534).[289] A good is non-rival if a good can be consumed by one individual without detracting the consumption opportunity to others from the same unit of the good (Cornes & Sandler 1996: 6). Jointness of supply is a property related to the cost or production function of a good (Mueller 2003: 11). If there is pure jointness of supply the marginal cost of extending consumption of a given good to an additional consumer is zero (Snidal 1979: 535).

It is often taken for granted that the assumptions of non-rivalry in consumption and non-excludability are completely satisfied (respectively completely not satisfied) and that there are only *pure* public and common goods (respectively pure private and club goods). However, next to these ideal types there are many goods for which the assumptions are only satisfied to some extent (Snidal 1979: 535). Even common examples for jointness of supply like national defense are not completely pure (Downs 1957: 16; Snidal 1979: 536). To protect a few citizens living abroad or in remote areas leads to extra expenses.[290] Drinking water is also not a pure collective good as sales of bottled waters in developing countries shows (UNESCO 2006: 402).[291] There is

[288] This difference in framing may influence individual's behavior (see also chapter 5.4.2.2).

[289] The proper wording is disputed although the differences are merely semantic (Head 1977: 228).

[290] Other examples are education, hospitals, and highways (Margolis 1955: 347).

[291] In cities of the developing world there is often either no municipal water supply at all or it fails to meet basic criteria of drinking water (UNESCO 2006: 403). Because of its availability and/or superior quality bottled water is in such cases traded as private goods. However, contrary to common belief it is also possible that the quality of bottled water is lower or far lower than municipal water. In the United States the *minimal standards* for bottled water, regulated by the FDA, are in fact lower than those for tap water that is regulated by the EPA (Gleick 2004: 31-32).

clearly rivalry in consumption for bottled water. According to Margolis (1955: 347) the crowded calendar of courts implies that its use by Alice makes the access to justice service less available to Bob.[292] Therefore, there is rivalry in consumption even for goods such as police and justice service. It is also quite clear that public roads get more crowded the more people drive on these roads (Downs 1957: 16). By using a public road Alice lowers the utility Bob derives from using the same road. For a public firework the addition of more viewers does foremost not detract the benefits enjoyed by others. However, as everybody knows from watching a firework this holds only true to a certain amount of viewers. If there are too many viewers the space gets overly crowded. Because of such crowding effects there is in reality for most public goods some degree of rivalry (Marwell & Oliver 1993: 42).

However, it may also be that additional users of a public good increase its value (see chapters 2.9.3 and 6). While some individuals may enjoy watching a firework alone most individuals probably prefer being accompanied by a small crowd. The value of some goods may also be heavily based on network effects. In such a circumstance free riders who do not contribute to a public good may even be beneficial for individuals contributing to it.

It is sometimes thought that non-rivalrous goods are automatically non-excludable (Taylor 1987: 7). This is not necessarily the case. Bridges and public parks are non-rivalrous as long as they are not overcrowded. However, for these examples there is clearly the possibility to exclude users and charge bridge tolls or park admission fees (Taylor 1987: 6).

Non-excludability

A characteristic of collective goods is *excludability* of benefits or non-control over excludability of benefits.[293] Goods for which benefits can be withheld costless display an excludable characteristic. Goods whose benefits are available to all as soon as they are provided are non-excludable (Snidal 1979: 539).

Similar to the argumentation made for non-rivalry there are not many goods that show pure non-excludability (Snidal 1979: 535). The problem of non-excludability is heavily related to the situation in which it would be theoretically possible to exclude others but in which the corresponding costs for excluding beneficiaries respectively for collecting fees were to rise above the revenues (Olson 1995: 14; Vanzandt 1993: 51-53). According to Stevenson (1991: 54) exclusion from benefits can be enforced for almost any good if very high costs are accepted. Costs to exclude free riders must be reasonable which is the case “whenever the gains in allocative efficiency … are greater than the associate costs” (Cornes & Sandler 1996: 3). Exclusion from collective goods is therefore sometimes just impractical respectively it does not pay off (Heckathorn 1996: 253; Mueller 2003: 11). One could for example imagine a technical solution applicable even centuries ago to exclude ships from benefiting lighthouse service (Vanzandt 1993: 52).[294] A lighthouse operator could place employees in illuminated boats along the outer range of the lighthouse.[295] When a ship approaches one of these boats could offer lighthouse service. If the fee is collected the small boat could ask the lighthouse to start

[292] In cryptography and physics a given set of names are taken for illustrating examples (Rivest, Shamir, & Adleman 1978; Schneier 1996: 23). Instead of using individual A or B the names Alice and Bob are used.

[293] For a discussion on the differences between excludability and control over excludability see Snidal (1979).

[294] For a further discussion concerning the famous lighthouse example see chapter 5.3.

[295] That this would have been theoretically possible can be seen by the existence of lightships (see Lenz 1960). However, these lightships did not collect fees for lighthouse service but acted as mobile lighthouses.

working by sending a secret sequence of light signals. However, although excludability for lighthouse service were theoretically possible, it would be hardly worthwhile. The lighthouse example shows in addition that the costs of excludability rest heavily on available technology. If ships and lighthouse operators had have radio (but not radar), the ship would have been able to radio the lighthouse operator to turn the light on in exchange for the audio-recorded promise to pay the fee. Probably the most difficult to exclude are ideas and trends. Once orange is set as the fashion color of the next year everybody can follow it. However, copyright, patent law, and the like may introduce some means for exclusion.

An additional aspect of excludability is law. It may prevent or enhance the possibility of free riding. In the lighthouse example the "Mare Liberum" or "Free Seas" principle (see Grotius 1978) made it illegal to force somebody paying fees (Vanzandt 1993: 51). As technology determines cost heavily, politics determines law. The "Free Seas" principle rested, at least superficially, on the idea of the benefits of free trade. For some public goods it is technically possible to exclude free riders at reasonable costs but it is politically not desirable. It is technically but not politically possible to exclude citizens in detached houses who do not pay taxes from fire department services. Health care systems are in fact in some countries a public and in others countries a private good.[296] A highway can be made either a toll road or a free road. Security can be enhanced by increasing police patrols that is primarily a public good or by installing additional door locks that are private goods. There are therefore some goods or services that are per se neither public nor private (Head & Shoup 1969: 567). A very good example that the degree of non-excludability depends on intention is software. Whether certain individuals are legally excluded or not rests on purposeful decision and not on technical possibilities. In the case of FOSS applications copyright holders impose consciously the legal restriction of non-excludability on their software.[297]

Snidal (1979: 539) observes that the exclusion from goods is central to the understanding of the political aspects of public good analysis. It can be concluded that excludability is largely a function of social institutions in interaction with technological features (Snidal 1979: 544) and political choice.[298] If discussing public goods the political and social context should therefore be considered (Snidal 1979: 558).

[296] A comparison between the United States, Switzerland, and Sweden shows that health care assistance may be a private, a partly private, or a public good. The health care system in the United States relies primarily on private insurance. In consequence on the average 16% of the U.S. citizens and 34% of Hispanics were uninsured in the year 2000 (Centers for Medicare & Medicaid Services 2005). In contrast to the United States it is in Switzerland mandatory to join a private but standardized basic health insurance (Schweizerische Eidgenossenschaft 1994, §3(1)). The Swiss cantons subsidizes basic health insurances for individuals with low or medium income with *at least* 50% (Schweizerische Eidgenossenschaft 1994, §65(1bis)). Supplementary health insurance is a private good that covers comfort (e.g. single room), dental treatment, alternative medicine (e.g. homeopathy), stays in a health spa, and the like. In Sweden the "financing of health services for the entire population is a public sector responsibility" (Swedish Institute 2003: 1). Roughly 80% of the costs are paid by county councils. The remaining 20% is almost entirely paid by the central government (Swedish Institute: 3). Patient fees are minuscule and nobody can be excluded from health care benefits.

[297] The copyright holder has the right to put his software under as many different license schemes as he wishes. This includes licenses that exclude users who do not pay fees.

[298] Hirschman (1970) argues that individuals may respond to organizational decline either with exit (withdraw from relationship) or voice (attempt to improve the relationship). It could be argued that exit is associated with solving the problem by relying on private goods and voice is associated with collective goods. An individual may respond to an outburst of crime by exit respectively by investing scare resources to move to a safer (and more expensive) area. However, he may also invest scarce resources to lobby for crime prevention such as having a social worker installed which is a public good. In many cases benefits from collective goods could also be obtained privately (Hardin 1982: 73). In such cases there are private goods that substitute collective goods. Excludability is then not only a function of social institutions or political decisions but also of individual decisions.

A thorough analysis reveals that there are not many *pure* collective goods (Snidal 1979: 544). Collective goods have some degree of collectiveness' ranging from zero to 100 per cent (Buchanan 1967: 11).

The lighthouse: Public or private good?

As discussed above it is not always easy to determine whether goods are collective or private. An interesting discussion emerged about the lighthouse, the classical example for a public good. Coase (1974) argues that the often cited lighthouse is not a good example for a public good as there were many privately owned lighthouses in England. The lighthouse owners used port fees for maintaining lighthouse services. Coase (1974: 375) concludes that "lighthouses were built, operated, financed, and owned by private individuals … The role of the government was limited to the establishment and enforcement of property rights in the lighthouse".

Private provision of lighthouses implies that the role of government is limited to enforcing property entitlements and freely agreed contracts (Vanzandt 1993: 54). Vanzandt argues that "under such an institutional scheme, lighthouse services will be provided spontaneously by the private decisions of ship captains and lighthouse operators". However, government support for lighthouse owners included in fact granted monopolies and government enforced fee collection, government provision of user taxes, or government provision of general taxes. With the exception of lighthouses run by religious persons or entities there were no private lighthouse owners who relied only on governmental protection of property and on governmental enforcing of consensual deals. Because of such heavy government involvement it seems to be questionable to assume that lighthouses in England were de facto private goods. The solution to the social dilemma was in fact not private ownership but a Leviathan that forced either users or the general public to pay for the lighthouses. This institutional setting seems to be very similar to the situation in that lighthouses are run by the government and subsidized with tax dollars. In consequence, the claim of Coase that lighthouses in England were not a public good seems to be wrong.[299]

The aggregation of preferences: The Condorcet paradox

In democratically organized societies individuals must participate in some sort of public decision making (Buchanan 1967: 4). Individual preferences must be aggregated. A special problem in the aggregation of individual preferences is noted by Condorcet (1785). He showed that collective preferences can be cyclic even if individual voter preferences are not. Let us for example assume that there is a community of three voters (Alice, Bob, Carol) who have to choose from three alternative modes of social action (1, 2, 3). The three individuals have the following preferences (listed in decreasing order of preference, adopted from Arrow 1950: 329):

- Alice: 1, 2, 3
- Bob: 2, 3, 1
- Carol: 3, 1, 2

In this example a majority of individuals prefers 1 to 2, 2 to 3, and 3 to 1. This result is a paradox because the requests of the majority are in conflict with each other. These conflicting majorities are made up of different

[299] Even if the argument of Coase (1974) were to hold true it would anyway largely be limited to England (Vanzandt 1993: 64).

groups of voters. Arrow (1950: 342) demonstrates with his possibility theorem that no voting system meets the following criteria if there are three or more choices:

i. Universality (334-336)
ii. Positive association of social and individual values (336-337)
iii. Independence of irrelevant alternatives (337-338)
iv. Citizen sovereignty (338-339)
v. Nondictatorship (339)

There is no method of voting which will remove this paradox (342). The consequence of Arrow's theorem is that aggregating individual preferences is at least theoretically non-trivial. Therefore, to the problem of providing a public good or preserving a common good the problem of preference aggregation is added.

5.4 Factors influencing collective action

There is a problem to get people to cooperate to overcome their “own biological limitations” (Barnard 1938: 60) which delimits individuals in their creative power. It is therefore necessary to analyze how and why individual choices are combined to produce collective outcomes respectively how and why they participate in collective decision making (Buchanan 1967: 3).

In modern societies decisions about collective action differ from the foretime. Cooperation cannot rely anymore only on family or neighbor ties; cooperation with strangers is compulsive (Marshall 1910: 6-7).[300] During history connections to neighbors and relatives were more and more abandoned in favor of connections to strangers. This reduces the importance of kinship and neighborhood as an explanation for the empirical observation of cooperation between individuals. In consequence, a wide range of other factors influencing cooperative behavior must be discussed.

That research on cooperative behavior is needed seems to be clear. Evidence from innumerable experiments show that people cooperate much more than it would be expected from pure self-interest (Rabin 1993: 1283). Dawes et al. (1988: 189) report that in one-shot public good experiments in which individually optimal contribution is zero or almost zero the contribution rate ranges between 40% and 60% of the socially optimal level. Hirshleifer (1985: 55) concludes that in economic theory:

> *… the analytically uncomfortable (though humanly gratifying) fact remains: from the most primitive to the most advanced societies, a higher degree of cooperation takes place than can be explained … for egoistic man.*

From these results it can be concluded that economic theory and self-interest is not sufficient for explaining behavior.

For Poppe (2005: 432) the essence of social dilemmas is specified by:

---

[300] An interesting variant of altruism that is not restricted to kinship is reported by Weinshtein (2005: 49-58). If a Tuvan nomad has hunted game it is completely shared within the village community. The hunter is the very last who receives his share. Interestingly, the game is not only shared within the village community but also with complete strangers. There is a tradition called Usha that urges to give a complete stranger a considerable part of the meet, the fur and the bowels if he meets the hunter. If there are many strangers, the hunter himself may go away empty-handed.

- the *context*,
- its *payoff structure*,
- its *decision structure* (outcome is high or low, certain or probable) and its
- *social structure* (large or small group, with or without communication, with or without leader).

For Messick (2000: 231-232) a useful framework analyzing real world social dilemmas should acknowledge at least three principles:

- Behavior is highly context dependant
- Much behavior is automatic, habitual, or rule following
- People differ with regard to how they interpret a situation and how they decide

Kopelman et al. (2002: 116) distinguish between *individual factors* (social motives and gender) and situational factors that influence cooperative behavior. Situational factors are further divided in *perceptual factors* (causes and frames) and *task structure*.[301] The task structure is broken down to *decision structure* (payoff and uncertainty) and *social structure* (power, communication, and group size).

The following discussion on factors influencing cooperative behavior is based on Kopelman et al.'s (2002) classification presented above. However, individual factors are complemented by *age*, *personal history,* and *emotions* (see Figure 54*)*. The decision structure is supplemented by *self-efficacy*, *decision sequence*, *response options*, and *contribution divisibility*. The social structure is endorsed by *group dynamics*, *fairness*, *leadership,* and *culture*. Finally, perceptual factors are complemented by *context*.

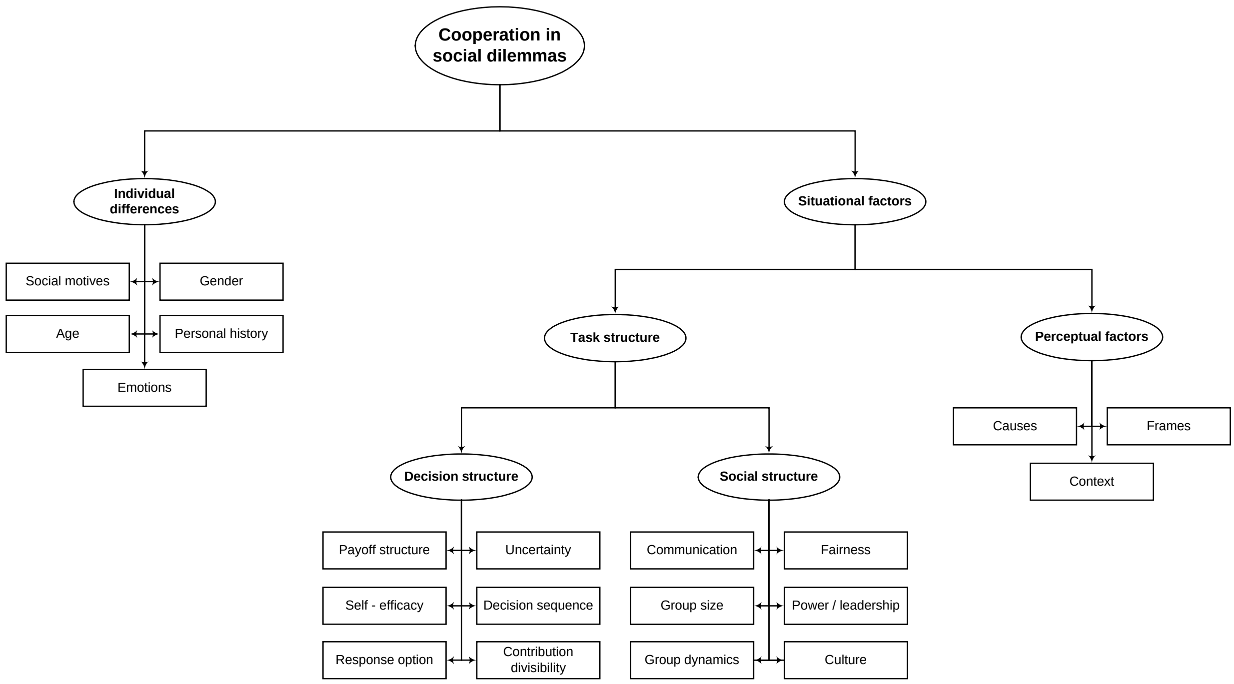


Figure 54: Factors influencing cooperative behavior[302]

[301] See also Weber et al. (2004: 287).
[302] Based on Kopelman et al. (2002: 116).

There are many different factors influencing cooperative behavior). Generally, laboratory experiments are designed to control subjects' *incentives* and limit *social* respectively *cultural* influences (Andreoni 1995a: 900). Experiments eliminate thereby on the one hand a large amount of subjects' natural tendency to be cooperative. On the other hand they provide little practical information to enhance cooperation. In the real world social and cultural factors can hardly be held constant. Knowledge about a factor fostering cooperation is of no use if in the real world this factor is *moderated* or *mediated* by other factors.[303]

However, on the other hand experiments that control for certain factors have the advantage to limit complexity. The interaction of an indefinite number of factors influencing cooperation cannot be handled. According to Biel (2000: 25) real life problems provide the proper structure of social dilemmas while experiments answer the circumstances under which individuals cooperate.

In the following, it is acknowledged that the interaction of different factors may influence cooperation. At the same time, the interaction of these factors is barely discussed.

5.4.1 Individual factors

Social motives

Cooperation theory (Deutsch 1949; Deutsch 1973) as well as dual concern theory (Blake & Mouton 1964; Pruitt & Rubin 1986) suggest that individuals have different social motives and that this influences behavior (De Dreu, Weingart, & Kwon 2000).[304] This contrasts most economic analysis; almost all game theory is based on the assumption that behavior is stable, well defined, and selfish (Dawes & Thaler 1988: 187). Experimental results frequently show that subjects do not maximize their own monetary payoff when others' payoff is affected (Charness & Rabin 2002: 817).

Models of social preferences assume that people are self-interested but that they are also concerned about the payoffs of others. Charness and Rabin (2002: 817-818) subclassified models of social preferences in i) difference aversion models, ii) social welfare models, and iii) reciprocity models. In difference aversion models players are motivated to reduce differences between others' and their own payoff. In social welfare models human beings quest for increasing social surplus with special care for those with low payoffs. Reciprocity models assume that individuals like to raise or lower others' payoff depending on how cooperative others were (for as discussion of fairness models see 5.4.2.1.2). In a series of experiments Charness and Rabin (2002: 817) found that individuals are more concerned with increasing social welfare than with reducing differences in payoffs. They found additionally that individuals are motivated by reciprocity as well.

Concerning social motives a very often cited classification is the one of McClintock (1972).[305] He argues that for motivation two dimensions are relevant: "own outcome" and "other's outcome". Based on these dimension he dissects four relevant social motives:

---

[303] A moderator is a qualitative or quantitative variable that affects the *direction* and/or *strength* of the relation between an independent and a dependent variable (Baron & Kenny 1986: 1174). A mediator variable *accounts* for the relation between an independent and a dependent variable (Baron & Kenny 1986: 1176).

[304] In cooperation theory the motivational orientations are a continuum ranging from cooperation to competition (Schei 2004: 4-5). In dual concern theory there are in contrast two separate dimensions for proself and pro-social behavior.

[305] See also Margolis (2005).

i. *Individualism* which is the motivation to maximize the own outcome

ii. *Competition* which is the motivation to maximize relative gains or the difference between one's outcome and the one of others

iii. *Cooperation* which is the motivation to maximize joint outcome

iv. *Altruism* which is the motivation to maximize others outcome[306]

Individualistic and competitive behavior is sometimes collapsed into pro-self behavior while cooperative and altruistic behavior is subsumed as pro-social behavior (Kopelman, Weber, & Messick 2002: 118).

In a comparison of different experiments on motives in social dilemma situations (Liebrand & Van Run 1985: 90) subjects show an insightful distribution concerning McClintock's motives (see Figure 55). Based on these results it can be concluded that pure altruism is by far less common than cooperation. This conclusion is backed by the observation that in one shot public good experiments contribution is typically between 40% and 60% (Dawes & Thaler 1988: 189).[307] Individualism seems to be somewhat more important than competition. Pro-self and pro-social behavior seems to be roughly equally important.

Social motives in social dilemmas

Subject's motive	Budescu et al. (1997)[1]	Kuhlmann (1975a)[1]	Kuhlmann (1975b)[1]	Kuhlmann (1976)[1]	Liebrand (1981)[2]	Liebrand (1983)[2]	Poppe (1980)[1]	Studies in US	Studies in NL	All 7 studies
Individualism	32%	23%	26%	26%	27%	29%	11%	25%	28%	26%
Competition	13%	28%	21%	18%	10%	10%	12%	19%	10%	17%
Altruistic	2%	9%	11%	11%	4%	5%	18%	9%	4%	8%
Cooperation	47%	17%	28%	30%	45%	52%	15%	29%	49%	34%
Other	6%	23%	14%	15%	14%	4%	44%	19%	9%	16%
N	205	205	167	128	132	122	98	705	254	959

[1] Conducted in the United States of America (US)
[2] Conducted in the Netherlands (NL)

Figure 55: Percentage of subjects classified with the corresponding motive[308]

[306] Altruistic behavior is not restricted to human beings. A sequence of experiments shows that chimpanzees help repeatedly toward humans and genetically unrelated conspecifics even if there was no direct or indirect reward and if helping were costly (Warneken et al. 2007: 1414). The degree of help towards unfamiliar humans by chimpanzees was similar to help by human infants.

[307] The arithmetic mean as a measure of central tendency is the sum of all measurements divided by the number of observations. Therefore, it could be possible that a mean of 50% is the result of one subject contributing 100% and one subject nothing. However, experimental results from Henrich et al. (2005: 803) suggest that is not case.

[308] There is some indication that the observed cross-cultural differences are related to different experimental settings and recruitment procedures (Liebrand & Van Run 1985: 90, 99-101).

In an interesting experiment Kuhlman and Marshello (1975) investigated the impact of three programmed strategies (always cooperation, tit for tat, and always defection) on the cooperation level in an iterated prisoner's dilemma as a function of three different motivational orientations (individualism, competition, and cooperation). After playing an iterated prisoner's dilemma game against human beings the subjects played in a second part of the experiment against one of the three programmed strategies. Before starting the game participants filled out a questionnaire for determining their motivational orientation. The authors expected, based on theoretical reflection, the following responses depending on motivational orientation and pre-programmed strategies (see Figure 56):

Motivational orientation and opponent strategy: Expected response

		Opponent's strategy		
		Always cooperating	*Tit for tat*	*Always defecting*
Subject's motive	*Individualism*	Defection	Cooperation	Defection
	Competition	Defection	Defection	Defection
	Cooperation	Cooperation	Cooperation	Defection

Figure 56: Expected response as a function of motivation and opponent strategy[309]

The relationships between motives, opponent's strategy, and cooperation level were similar as suggested (see Figure 57).

Motivational orientation and opponent strategy: Experimental response

		Opponent's strategy		
		Always cooperating	*Tit for tat*	*Always defecting*
Subject's motive	*Individualism*	31%	***63%***	13%
	Competition	14%	15%	8%
	Cooperation	***92%***	***80%***	29%

Figure 57: Proportion of cooperative choice in the motive x strategy interaction[310]

Participants with a *competitive* motivation hardly cooperated irrespective off the opponent's strategy. For *individualistic* participants it was in the long term only rational to be cooperative if the opponent played tit for

[309] Kuhlman and Marshello (1975: 926).
[310] Kuhlman and Marshello (1975: 929).

tat. If the opponent always cooperates or defects it is better to always defect. In fact, individualistic participants cooperated considerably higher if the opponent played tit for tat. Participants with *cooperative* motivation have chosen to respond positively to the opponents' tit for tat strategy because this maximizes personal as well joint benefits. However, in contrast to individualistic players they did not exploited altruistic opponents (cooperate always). This behavior maximizes joint but not personal outcome of the cooperative participants. Cooperative individuals reacted with defection to the strategy of always defecting.

The above presented results are corroborated by other experimental results (Kramer, Mcclintock, & Messick 1986; Liebrand 1984; Liebrand & Van Run 1985; Roch & Samuelson 1997) as well as survey studies (Van Vugt, Meertens, & van Lange 1995; Van Vugt, van Lange, & Meertens 1996).[311] In a resource dilemma experiment of Budescu et al. (1997: 189) competitive subjects harvested the most, followed by individualistic subjects, cooperative subjects, and finally altruistic subjects. In another experiment competitive individuals harvested more than individualistic individuals who in turn harvested more than cooperative individuals while altruistic individuals harvested least (Liebrand 1984). However, in some studies the relationship between motives and decision making was moderated by uncertainty (Roch & Samuelson 1997) or knowledge about resource debility (Kramer, Mcclintock, & Messick 1986). It seems therefore reasonable to assume that individuals with competitive or individualistic motives display in general higher non-cooperative behavior than the ones with altruistic or cooperative motives.[312]

In an interesting approach Fischbacher and Gächter (2006) tested whether actors have different social preferences and whether they act consistently. Based on a one-shot social dilemma created by Fischbacher et al. (2001) the authors researched contribution preferences.[313] They found that an absolute majority of 55% of individuals are conditional cooperators, that 23% are free riders while the rest show a more complicated pattern of behavior (Fischbacher & Gächter 2006: 13). In a second standard social dilemma game with 10 periods Fischbacher and Gächter researched whether behavior is consistent with the expressed preferences in the first social dilemma game. In fact, concerning their preferences individuals were overall very consistent (Fischbacher & Gächter 2006: 22).

There are probably very few situations that induce purely cooperative or competitive behavior (De Dreu, Weingart, & Kwon 2000: 889; Deutsch 1949: 132). Most everyday situations include a multifaceted set of goals and sub-goals. In some situations individuals compete and in other situations they cooperate with each other. Deutsch (1949; see also Luschen 1970) presents the examples of a basketball team in which the players cooperate to win the game but compete to be the star of the team. Correspondingly, different cigarettes companies cooperate with respect to the goal of increasing the general consumption but compete in respect to the maximization of sales of their own cigarette brand.

---

[311] In these survey studies (Van Vugt, Meertens, & van Lange 1995; Van Vugt, van Lange, & Meertens 1996) not only motives but also behavior (cooperation vs. defection) is self reported. This procedure has potentially a methodological problem. The possibility of favorable self-presentation cannot be excluded. Although it is better to measure behavior instead of relying on reported behavior the matching of the results with the ones of experimental studies suggest that the there is no considerable effect of favorable self-presentation.

[312] In some experimental studies individualistic and competitive motives are combined and labeled together as noncooperation motives while cooperative and altruistic motives are classified together as cooperative motives (Roch & Samuelson 1997: 223).

[313] Subjects had to specify how much they want to contribute to a public good for every average contribution level of other group members.

Based on cooperative motivation corresponding norms may evolve. One of these norms may be conditional cooperation in which one cooperates if the others cooperate as well. Because others cooperate individuals may infer that there is such a norm and comply. This belief may be explained by the observation that it is hard to fake cooperation successfully in the long run (Dawes & Thaler 1988: 192).[314] However, conditional cooperation is only possible if individuals act sequentially and if they know what others did in the past (Van der Heijden et al. 1998: 6).

Emotions, passion, and mood

Hume (1739: 415) noticed that reason "alone can never produce any action" and that it is the slave of passion.[315] Indeed, pro-social emotions seem to play an important role in cooperative relations (Bowles & Gintis 2003). Emotions are defined by cognitive antecedents, intentional objects, physiological arousals, physiological expressions, valence, and action tendencies (Elster 1998: 49). Although these characteristics of emotions cannot be discussed in length it should be noted that emotions differ from visceral factors such as pain or hunger in that they are triggered by beliefs (cognitive antecedents). Thereby emotions are related to social motives already discussed above. In contrast to visceral factors emotions also do have an intentional object that means that they are *about* something. There are different kinds of emotions (Elster 1998: 48):[316]

- Social emotions such as anger, hatred, guilt, shame, pride, admiration, and liking
- Counterfactual emotions such as regret, rejoicing, disappointment, and elation
- Emotions about what might happen such as fear and hope
- Emotions about what happened such as joy and grief
- Emotions triggered by other's possessions such as envy, malice, indignation, and jealousy
- Special emotions not building a coherent category such as contempt, disgust, and romantic love

Research on mood found that a positive mood increases the willingness to show pro-social respectively helping behavior (Salovey, Mayer, & Rosenhan 1991; Schaller & Cialdini 1990). Others also suggest a relation between positive mood and cooperative behavior (Barry & Oliver 1996; Forgas 1998; Isen & Baron 1991; Oatley & Jenkins 1996). However, Hertel et al. (2000: 443) claim that these studies may have methodological flaws. In some of the studies mood states could have affected the comprehension of the experimental task. Manipulation of negative or positive mood may also have an influence on other concepts relevant to cooperation such as injustice. Additionally, Hertel et al. (2000: 443) identify some theoretical problems. The connection between cooperation and mood for example is not always clear. It may be that uncooperative behavior is associated with positive affects such as successful competition. In their own studies Hertel et al. (2000: 465) found that a positive mood leads to a more heuristic decision style while a negative mood leads to a time-consuming and

[314] Dawes and Thaler (1988: 192) quote Senator Sam Ervin who said that the "… problem with lying is that you have to have a perfect memory for what you said" and stated that is easier to remember what actually happened.

[315] While emotions are a mental state a mood is the combination of different emotions. A passion is a strong positive emotion. Yet there are definitional differences it is not intentionally distinguished between emotion, passion, and mood in the following.

[316] Whether surprise, boredom, interest, sexual desire, enjoyment, worry, and frustration qualify as emotions is not clear (Elster 1998: 48).

systematic decision style. Because the decision style had a connection to cooperation mood served as a moderator.

A series of studies conducted by Stouten et al. (2006) showed that emotional and retributive reactions to a violation of the equality rule in social dilemmas are a function of trust. High trusters for example want to know why an equality rule is violated (Stouten, De Cremer, & van Dijk 2006: 903). Depending on the cause high trusters in contrast to low trusters reacted differently. The authors also found that in high trust situations subjects reacted more emotionally to the violator and were more willing to engage in retributive reactions.

Emotions such as anger play an important role in the punishment of free riders (Bosman & van Winden 2002; De Quervain et al. 2004; Hopfensitz & Reuben 2005). The effectiveness of punishment, however, does to some extent depend on shame and guilt (Hopfensitz & Reuben 2005: 1).[317] If punished individuals do not feel guilty and react with anger as well multiple rounds of punishment and retaliation are introduced which leads to a destruction of the collective good.

Emotions are not relevant in all situations. Relations among husbands, parents, and children are more likely to be guided by emotions such as love and guilt than relations among employees of a firm (Becker 1996: 151). Therefore, the importance of emotions depends on the situation.

Gender

Although a considerable amount of research has been dedicated to the question whether gender influences behavior in social dilemmas no conclusion has been reached (Walters, Stuhlmacher, & Meyer 1998: 4). Some studies have found female to be more competitive than male (Conrath 1972; Fisher & Smith 1969; Kimmel et al. 1980; Tedeschi, Bonoma, & Lindskol.S 1970). Other results suggest the opposite (Bedell & Sistrunk 1973; Hottes & Kahn 1974) or that there is no relationship (Ferguson & Schmitt 1988; Grant & Sermat 1969; Horai & Tedeschi 1975; Wall & Blum 1991; Watson & Hoffman 1996). In the meta-analysis of Walters et al. (1998) female are more cooperative. However, gender could only explain l*ess than one percent* of the variance in behavior.

Based on these results it seems to be reasonable to assume that there is either no gender effect in social dilemmas or that it is completely negligible.

Age

By playing anonymous dictator and ultimatum games with children of different school grades Harbaugh et al. (2002) could show that age does have an influence on pro-social behavior.[318/319] Children playing the dictator game make considerably smaller proposals than adults do (Harbaugh, Krause, & Liday 2002: 28). They also make and accept smaller ultimatum proposals than adults.

---

317 In repeated games in which players meet again self-interest may also demand to react cooperatively after a punishment in order to pretend good faith. However, in situations in which the same players do meet only a few times and in which there is no reputation system self-interest is not a good explanation for a cooperative move after being punished.

318 The game protocols were adapted to make them understandable to children. The differences between the dictator and the ultimatum game indicate that the children seem to have understood the implications of the different rules.

319 With some children the dictator game was played first and the ultimatum game second and vice versa.

The youngest children make in the dictator as well as in the ultimatum game the lowest proposals (see Figure 58). While second graders offer in the dictator game on the average only 4% of their tokens, the twelve graders offer more than one fifth of their tokens. To some extent the same pattern can also be observed for the ultimatum game.

		Dictator game	Ultimatum game	
Grade	N	Mean offered	Mean offered	Rejection rate
2	74	4%	35%	11%
4	106	14%	41%	14%
9	90	12%	45%	3%
12	40	21%	43%	10%

Figure 58: Dictator and ultimatum game proposals and rejection rates by grade[320]

Personal history

It is well known that the shadow of the future affects cooperative behavior (Axelrod 1984). However, individuals are also influenced by past experience *with unrelated other subjects*[321] respectively by the shadow of the past although from a rational point of view the past is in this case irrelevant (Gautschi 2000: 131 and 141).[322/323] It is thought that the importance of personal history can be attributed to social learning (Weber, Kopelman, & Messick 2004: 289).

Bettenhausen and Murnighan (1985) demonstrated in an iterated prisoner's dilemma the effect of personal history on cooperative behavior. In advance to the game half of the subjects played in two practice sessions the iterated prisoner's dilemma in which the fictional opponent behaved cooperatively. In addition, the group was allowed to discuss with each other in advance to the two practice sessions. A confederate presented some arguments for behaving cooperatively. In the other group the fictional opponent behaved competitively in the practice sessions and a confederate provided rationales why to behave competitively. After this practice sessions the subjects with the exception of the confederates were regrouped. These groups were either homogenous or heterogeneous concerning their practice session experience. In homogenous groups both subjects had either a cooperative or a competitive experience. In heterogeneous groups there were subjects with a cooperative and subjects with a competitive experience. In the experiments itself there was no other difference between the groups than the personal history of the subjects. The results of the experiment showed that homogenous pairs with cooperative experience were significantly more cooperative than pairs with a competitive experience (Bettenhausen & Murnighan 1985: 32).

---

[320] Harbaugh et al. (2002: 11).

[321] Certainly, situations in which the very same subjects meet again are different. In this case a tit for tat reciprocity may develop.

[322] For a discussion of the influence of positive affects on intrinsic motivation, extrinsic motivation, and self-determination theory see Isen and Reeve (2005).

[323] Past experience influences decision making beyond social dilemmas. An interesting empirical study (Post et al. 2007) researched the television show „Deal or no deal" aired in almost 50 countries from Argentina to Zimbabwe (Wikipedia 2007a). The contestants have to make repeated risky choices and may win considerable amounts of money. In the Dutch version of the game the winnings range between 0.01€ and 5'000'000€ (Post et al. 2007: 2). Playing the game does not require special skills, knowledge, or strategy. The amount of money won depends basically on luck and risk aversion. The structure of the decision making problem does not change in the different rounds of the game. Post et al. (Post et al. 2007) have shown that decision making of the contestants is considerably dependent on the outcomes experienced in the previous rounds of the game.

That past positive experience increases future cooperative behavior towards unknown other individuals is shown by the following experiment. Individuals who found a coin in the coin return of a public phone were more likely to help a stranger to pick up papers that had been dropped than individuals in the control group (Isen 1987, cited in Rutte & Taborsky 2007: 1421). Rutte and Taborsky (2007) found a similar relationship in an experiment with wild-type Norway rats. Anonymous receipt of help increased the propensity to help by more than 20% (Rutte & Taborsky 2007: 1422). Such mechanisms of generalized reciprocity[324] have the advantage over other types of reciprocity (see chapter 5.5.1.3) that less cognitive abilities are needed (Rutte & Taborsky 2007: 1421). Individuals only need to remember the past experience. They do for example not need to recognize each other.

5.4.2 Situational factors

5.4.2.1 Task structure

5.4.2.1.1 Decision structure

Payoff structure

Experimental research on social dilemmas demonstrated significant effects of different payoff structures (Kopelman, Weber, & Messick 2002: 122). There is a tremendous amount of games with divergent payoff structures (Gintis 2000; Rasmusen 1989). Some of the more known games are the battle of the sexes, the chicken game, the coordination game, the dictator game, the stag hunt game, and the ultimatum game. The most known game is however the prisoner‘s dilemma (Kuhlman & Marshello 1975: 922).

Each of the above mentioned games have a different payoff structure that affects behavior. A perfectly rational egoist does not cooperate at all in a dictator game and does not share any money with other individuals (Bolton, Katok, & Zwick 1998: 270). In coordination games like for example the stag hunt game (see below) the same rational egoist clearly prefers to cooperate. The problem here is to learn about the choice of the other player. The difference between the dictator and the stag hunt game is that in the former game incentives for defection are, in relation to the incentives for cooperation, higher than in the later game and vice versa.

Within rational choice models increasing relative incentives for cooperation generally increases cooperative behavior (Weber, Kopelman, & Messick 2004: 290). This may also have influences beyond self-interest. Kirchgässner (1992) argued that in low-cost decision situations soft incentives such as moral rules can have a much stronger impact than economic hard incentives. It can therefore be inferred that individuals stick more likely to social motives discussed in chapter 5.4.1 the lower the costs for such behavior are. This is also reflected in Margolis' (1984; 2004; 2005; 2006) **NSNX** principle that states that individuals do want **N**either to be **S**elfish **N**or to be e**X**ploited. In his model there is an equilibrium if $W = G' / S'$ where G' is the value of a marginal contribution to a group, S' the marginal value to self-interest, and W the weight given to self-interest. Whenever W is bigger than G' / S' an individual acts in his self-interest and whenever W is smaller than G' / S'

[324] Other names of generalized reciprocity are up-stream tit for tat reciprocity and up-stream indirect reciprocity (Rutte & Taborsky 2007: 1421).

in the group interest. In this NSNX model an individual acts in the group interest but only if it is not too expensive.[325]

Normally the discussion about the payoff structure refers to the payoff of the observed subject. This is correct for pro-self oriented individuals. However, for pro-social orientated individuals the payoff of others is also relevant (see also chapter 5.4.1). According to De Alessi (1975: 127) there is for example evidence that in the period following a disaster cooperative behavior increases (see also self-efficacy). Hirshleifer (1963: 6) states that in an immediate post-impact period of a disaster a strong feeling of community identification is generated which promotes cooperative and unselfish efforts toward repair and relief activities.[326] Outside the impact zone of the disaster a "counter-disaster-syndrome" can be observed that includes an outburst of generous assistance. In emergencies prices for food do often not increase enough to clear the market. This cooperative behavior seems to be at least partly motivated by altruistic concern. There are several reasons why helping behavior from outside and inside the disaster area cannot be solely attributed to an informal insurance fee as sometimes suggested (De Alessi 1975: 136). *Firstly*, there is some anonymous helping. *Secondly*, there is an enforcement problem and selfish beneficiaries of help can hardly be expected to reciprocate.[327] *Thirdly*, old, sick, and poor people also get help although they are unable or unlikely to reciprocate.

A problem observed in disasters is the destruction of leadership and public service (Hirshleifer 1963: 7). One reason for this is that leaders are in conflict between personal (S') and public (G') roles (see NSNX principle above). In consequence, "many leaders and essential personal … abandon their posts in order to see to their own families' safety". The incidences after the landfall of Hurricane Katrina on 29. August 2005 near the city of New Orleans (US) confirms this statement. Many police officers left their post that resulted very quickly in the destruction of order. This included looting and shooting, targeting rescue helicopters and relief workers. To restore order Governor Kathleen Blanco was forced to warn lawbreakers that the National Guard troops "hardened on the battlefield in Iraq had landed in New Orleans and had orders to shoot to kill …. These troops know how to shoot and kill, and they are more than willing to do so, and I expect they will" (Treaster & Sontag 2005: 1). The example of disasters shows that individuals take the own payoff and the one of others into account if they make a decision in a social dilemma.

[325] To illustrate the NSNX principle let us assume the following example (adapted from Margolis 2006: 16-17): An individual is watching the Super Bowl. By chance he sees that the neighbor's house caught fire. Let's further assume that i) the own house is far enough away so that the fire could not jump over and that ii) this individual will move very soon to another town so that the neighbor has hardly a chance to reciprocate. In this situation an individual will most likely help the neighbor to extinguish the fire. The utility from helping (G') is higher than watching TV (S'). However, if by chance the own house caught fire as well the individual will probably not help his neighbor but save his own house. While the utility from helping the neighbor (G') remains the same the utility from self-interested behavior respectively saving the own house (S') is higher. Let us assume a different variant of the situation in which the individual is watching the Super Bowl and the neighbor cuts his lawn. By chance the individual sees that the power mower stops working. Because he is in contrast to the neighbor an engineer he is quite sure to know how to repair it quickly. Whether the individual helps the neighbor depends on his weight he gives to self-interest (W) respectively how much he enjoys watching TV (S') and how much he thinks helping the neighbor (G') is a good thing to do.

[326] However, this effect gradually wears off after a couple of days or weeks (Hirshleifer 1963: 6).

[327] This is only true if a beneficiary does not assume the disaster to happen again. Otherwise a tit for tat reciprocity could emerge.

Self-efficacy

An important factor influencing cooperative behavior is self-efficacy (Komorita & Parks 1995: 193). Individuals cooperate more likely if they think their contribution is critical for the provision of the collective good (Komorita & Parks 1995: 193; Rapoport, Bornstein, & Erev 1989). In large groups individuals believe that their effort is insignificant (Kerr 1989; Rapoport 1985). Consequently, individuals in large groups may refrain from contributing. These findings are related but not totally in line with Olson's (1995: 49-50) arguments about privileged groups. In privileged groups at least one group member has enough incentives to provide the collective good even if he has to bear the full burden.

Andreoni (1995b: 12; see also Bagnoli & McKee 1991) found that cooperative behavior is enhanced for step-level public goods (see chapter 5.7). In this situation it is known to everybody that the public good will only be provided if there are enough contributions. Every single contribution may turn a potential public good from being useless to being fully beneficial. Therefore, even a minuscule contribution may be critical for the provision of the public good. For continuous public goods a minor contribution will not change its value dramatically.

There is also the observation that individuals often overcome the free rider problem in disasters (Harrison & Hirshleifer 1989: 203, see also payoff structure). Harrison and Hirshleifer call this situation a "weakest link environment" that is characterized by the fact that the failure of any unit may be fatal to the whole.[328][329] This refers to desperate circumstances in which everybody must do his duty if the community should survive. An example might be townspeople manning sectors of a dike when a river or the sea is in flood. One person's failure was to mean that the water would break through and flood the entire town. Such behavior may or may not be related to pro-social behavior.[330] If a collective good collapses without cooperative moves of everybody it is for rational egoists beneficial to cooperate (Harrison & Hirshleifer 1989: 203). However, a pro-social subject may also contribute even he is hardly affected by the outcome. He may do so because otherwise the good that has benefits for others will not be provided. In the river example an individual may do his best even if his house is save on a hill and nobody could ever detect afterwards which part of the dam collapsed first.

Behavior related to self-efficacy can also be observed in animal cooperative breeding. Cooperative breeding refers to situations in which adult individuals aid the genetic parents in the rearing of the young (Emlen 1991: 301). In a study on white-fronted bee-eaters Emlen (1982: 35-37) shows that helping behavior of this species rests heavily on the harshness of condition.[331] In the Rift Valley in Kenya where this bird lives there is a tremendous year-to-year variation in timing and amount of rainfall. Rainfall is an indicator for environmental harshness because it has an influence on insect population that is in turn the nutrition of the white-fronted bee-eater. If there are insects in abundance no helping is needed and vice versa. Observation shows that the

[328] Security and especially computer security is „a chain: the weakest link breaks it" (Schneier 2004: 272).

[329] The best-shot model is the opposite extreme (Harrison & Hirshleifer 1989: 204). In this case different teams engage in a competition in which victory benefits the whole team but the victory solely relies on the best individual performance within a group. A related example is an antimissile battery firing under local control at a single incoming missile.

[330] That the maintenance of dikes in the past was not only based on pro-social behavior but also on proper institutions is apparent. There were laws and courts solely responsible for dike maintenance (e.g. see Deichgerichtsbarkeit in Gabriel 2004: 112-114). That the maintenance of dikes was a serious issue can be seen by the Deichordnung des Herzogtums Bremen (dike law of the duchy Bremen): According to chapter XVI § 1 individuals who penetrated dikes with intent were to be executed by burning (Djuren 1963: 75).

[331] Helping other birds raising their offspring is costly (Emlen 1991: 301). Even if the conditions are so harsh that a bird decides not to attempt independent breeding it is easier doing nothing than helping other birds.

percentage of the bird population that helps other birds is significantly correlated to rainfall. It can be concluded that this species helps the more such behavior is needed.

The basic human need competence (Ryan & Deci 2000a: 58) is connected to self-efficacy. People want to succeed in making a difference if they contribute. At the same time success is a strong feedback indicating competence.

Uncertainty

According to Ostrom (1997: 33) the decisions related to common pool resources are usually uncertain. This position is backed by de Vries and Wilke (1992: 81) for whom uncertainty is quite common in real-life resource dilemmas. Conversely, in many experiments on social dilemmas subjects have complete information about the structure of the game and the payoff matrix (Gibbons 1992: 1).[332] Each player's payoff structure is then common knowledge. Unknown is often only the behavior of other individuals. However, in reality there are several sources for uncertainty (Ostrom 1997: 33-34):

- ***External influences:*** There are external sources of uncertainty like for example rainfall, temperature, or market prices.
- ***Structure:*** Uncertainty about the exact structure of the good such as its boundaries and internal characteristics.
- ***Consequences:*** How actions affect the collective good is often uncertain. It is immediately not apparent how much waste can be dumped into a lake before it is rendered unusable (Rapoport et al. 1992: 44).
- ***Behavior:*** Uncertainty about the behavior of individuals. The uncertainty whether others will cooperate or not is in fact an important characteristic of social dilemmas (Gustafsson, Biel, & Garling 1999: 48). This kind of uncertainty does occur if decisions are made simultaneously (De Vries & Wilke 1992: 84).

In some cases uncertainty about a collective good may decrease as a by-product of contribution. The using of a grazing range leads to knowledge about its internal structure. In other cases it needs major investments to decrease uncertainty. An example is the structure of groundwater basins that have to be analyzed by costly geological expertise. Although uncertainty can be reduced by investing scare resources total certainty can hardly be accomplished.

Uncertainty is not only a major characteristic of social dilemmas but it has also major and subtle effects on behavior (Weber, Kopelman, & Messick 2004: 291). In a resource dilemma there were two groups of subjects; in one group there was high uncertainty about the size of the resource and in the other low uncertainty (Gustafsson, Biel, & Garling 1999: 50-56). Subjects in the highly uncertain condition harvested more than the ones in the low uncertainty condition (Gustafsson, Biel, & Garling 1999: 55). Rapoport et al. (1992: 50) observes similarly that "under environmental uncertainty, subjects requests more from the resource, with the most dramatic increase in the case of extreme uncertainty". Gustafsson et al. (1999: 62) suggest "that the overharvesting effect of resource uncertainty in a resource-dilemma is a special case of an individual outcome-

[332] Complete information should not be confused with perfect information that includes knowledge about the entire decision history of the other subjects (Fudenberg & Tirole 1989: 264, 1991: 72; Gibbons 1992: 55).

desirability bias”. Individuals tend to think that the outcome in social dilemma will hit the spot anyway. Rapoport et al. (1992: 55) agree that optimism or wishful thinking partly explains non-cooperative behavior in social dilemmas. Another explanation is provided by de Vries and Wilke (1992: 81). They argue that the observation of overharvesting in uncertain situations can be explained by what they call constrained egoism. Subjects’ egoism is constrained by their preference for fairness and equal outcomes. If there is uncertainty about the size of the resource a fair share cannot be computed anymore. This makes the constraint ineffective and leads in consequence to an egoistic bias.

In an interesting experiment Roch and Samuelson (1997) investigate the effect of social value orientation, environmental uncertainty, and trials on harvesting from a replenishable common pool. In difference to Gustafsson et al. (1999) and Rapoport et al. (1992) the uncertainty was operationalized by the unknown rate of resource regeneration and not by the size of the initial common pool. As expected by Roch and Samuelson (1997: 230) non-cooperative subjects harvested significantly more than cooperative subjects (see also social motives in chapter 5.4.1). In accordance with the presented results above subjects harvested more in the highly uncertain situation. More interestingly, social motives moderated the effect of uncertainty on harvesting during the middle series of the totally ten trials.[333] Noncooperators responded to high uncertainty at this phase by increasing their harvest while cooperators continued using self-restraint (Roch & Samuelson 1997: 203). One may suggest that the cooperators responded to the (uncertain) threat of resource depletion by restraint while non-cooperators responded by increased harvesting to take out as much as possible before the resource is destroyed.

Frank Knight (1921 part 3, chapter 4, 36) distinguishes between three possibilities of probability judgment:[334]

i. ***A priori calculation*** is used in games of chance. An example is figuring out probability of throwing a certain number with a perfect die.

ii. ***Statistically calculation*** is reached by the empirical method of applying statistics to actual instances. Figuring out chances that a building will burn rests on the empirical proportion of similar buildings destroyed by fire in a given region and time.

iii. ***Estimates*** for which there is no valid basis for calculation. An example is the probability of having made a mistake in a unique decision, for example expanding work capacity. If one were to know the mistake, one could and would eliminate it.

For *a priori* and *statistical calculation risk* is the managerial problem. For *estimates* the problem is *uncertainty*. Situations in which a priori calculation is possible do practically not exist in business environment (Knight 1921 part 3, chapter 4, 26). Mostly estimates are relevant because there is no valid basis for calculation that “accounts for the peculiar income of the entrepreneur” (Knight 1921 part 3, chapter 4, 48). An experiment that is based on a priori calculation and not on statistical calculation or estimates is therefore unrealistic.[335]

---

[333] Based on the structure of the game the threat of depletion was only virulent at the middle of the trials. At the beginning it was not possible that the resource degenerated completely and at the end of the trials resource breakdown would have been no problem because the game was finished anyway.

[334] However, there “has always been a good deal of skepticism about the behavioral significance of Frank Knight's distinction” (Ellsberg 1961: 643).

[335] However, an unrealistic experiment is not necessarily a valueless experiment.

Contribution divisibility

In many experiments on social dilemmas contribution is discrete. The subjects have to donate all or no coins. This has a strange consequence for cooperative individuals (McClintock 1972): they are forced to behave either altruistically or individualistically. It could be guessed that in such situations with binary choices subjects will more often decide that their own outcome is more relevant than the one of others. If this reasoning is true cooperation rates may therefore increase if decisions can be made continuously (Komorita & Parks 1995: 195).

Continuous contribution has two important consequences: *Firstly*, it facilitates to start cooperatively in uncertain situations. An actor who is unsure about the intentions of the other players may donate a small amount of his tokens to the collective good. This decreases the risk of being exploited. If other actors defect not much is lost. However, if the other actors reciprocate the amount of cooperation can be increased and conditional cooperation may evolve. Similarly, an actor does hardly donate all his belonging in a real life social dilemma. However, donating 5$ may be a likely course of action to test whether others are cooperative or not. *Secondly*, actors whose motivation to cooperate is small can also contribute to a collective good. Let us assume that somebody asks an individual to donate for a good cause. This individual agrees that the cause is really a good one but does not care too much. If the only possibility is to donate 10'000$ the individual, if it is not a billionaire, will hardly make a donation. However, if he is free to give whatever he thinks is appropriate he will put 10 dollars in the collecting box. Continuous contribution is therefore very important for collective goods for which an underprovision or an overuse does not have very grave consequences.

Decision sequence

There are two different forms to describe the decision structure in game theory: an extensive and a normal (strategic) form (Fudenberg & Tirole 1989: 261). This distinction is crucial for understanding the importance of decision sequence in social dilemmas. The extensive form specifies not only the sequence of moves but also the informational conditions under which a game takes place (Dresher 1961: 74-76; Harrison & Hirshleifer 1989: 203). If decisions are made sequentially a succeeding player knows the decisions made by preceding players. It is argued in chapter 8 that the success of FOSS relies partially on sequential decision making. Therefore, the differences between the extensive and the normal form are discussed in length below.

Extensive form

The extensive form consists of (Fudenberg & Tirole 1991: 77; Gintis 2000: 10):[336]

i. the set of players,

ii. the order of moves,

iii. the players' payoff as a function of the moves that were made,

iv. what the players choices are when they move,

v. what each player knows when he makes his choices, and

[336] It is possible to use a mathematical notation to define an extensive game but this seems to be too laborious for a simple conceptual idea (Gintis 2000: 10).

vi. the probability distributions over any exogenous events.

The game tree represents the possible moves and their order. It is made up of a number of nodes connected by branches. Each node constitutes a decision point for a player that is specified above the node. The branches out of the node constitute the possible actions. The example of Figure 59 represents a sequential prisoner's dilemma.[337] Player 1 has to decide whether to cooperate (C) or to defect (D). Afterwards player 2 faces the same decision situation with the difference knowing what player 1 has chosen. The advantage of this representation is that dynamic interactions between players can be shown.

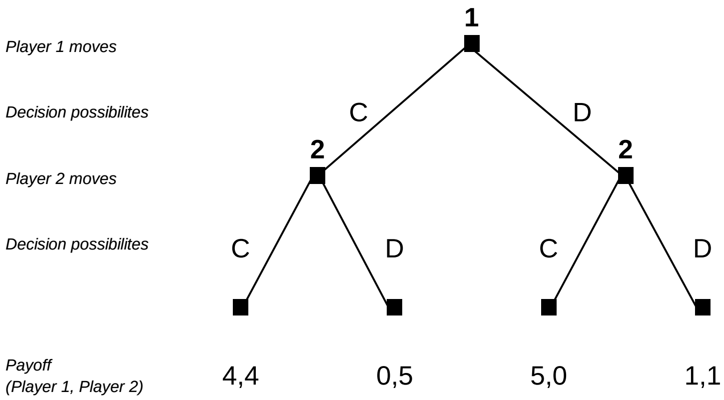


Figure 59: Extensive form of game representation

Normal form

In the normal form a game consists of a number of players, a set of strategies, and a payoff structure (Gintis 2000: 12). The details of the extensive form represented by the game tree are omitted which allows to concentrate on strategic aspects of a game. However, the dynamic structure of the game is neglected (Eichberger 1993: 2).

	Player 2 chooses C	Player 2 chooses D
Player 1 chooses C	4,4	0,5
Player 1 chooses D	5,0	1,1

Figure 60: Normal form of game representation

Social dilemma studies have mostly adopted a simultaneous protocol of play (Budescu, Au, & Chen 1997: 180; Erev & Rapoport 1990: 405). However, in reality many decisions are made sequentially. Especially for explaining herding behavior it is crucial to assume a sequential decision making model (see for example

[337] A simultaneous prisoner's dilemma is represented by a dotted line between the decision possibilities of player 2. In the case of the prisoner's dilemma the dominant strategy of a rational egoist does not depend on whether players decide simultaneously or sequentially (see also chapter 5.1). However, if individuals are prosocially motivated or if they have a preference for fairness the decision sequence matters.

Banerjee 1992). It may therefore be that research on some types of social dilemmas gives a wrong picture of cooperation (see Erev & Rapoport 1990: 422).

Protocol of play

To change the protocol of the game to sequential moves does not have any influence on defection of rational egoists if there is a dominant strategy like in the prisoner's dilemma. However, in some games like the stag hunt game (Skyrms 2003: 3) or big monkey and little monkey game (Gintis 2000: 3-9) cooperation rates may change dramatically (Budescu, Au, & Chen 1997; Erev & Rapoport 1990: 422; Harrison & Hirshleifer 1989: 205; Komorita & Parks 1995).[338] Games played in sequence include for example the centipede game (McKelvey & Palfrey 1992), the dollar auction game (Shubik 1971), the Kuhn poker (Kuhn 1950), the ultimatum game (Guth, Schmittberger, & Schwarze 1982), and the trust game (Burks, Carpenter, & Verhoogen 2003).

As van der Heijden et al. (1998: 6) argue a norm of reciprocity can only evolve if a game is played sequentially and if the reacting player is informed about the action of the other player. If not, the player does not know anything about the action of the other player that is a logical necessity for reciprocity respectively conditional cooperation. Erev and Rapoport (1990: 422) conclude that "the sequential protocol of play … is significantly more effective in solving the public good problem than the simultaneous protocol".

In an experiment on behavior in sequential games Budescu et al. (1997) played resource dilemmas with four distinct protocols:[339/340]

- Under the *simultaneous protocol* all subjects make their harvesting requests simultaneously.
- In the *sequential protocol* the subjects make requests one after the other with complete information about their position (5 positions) in the sequence order and full disclosure of previous requests.
- In the *positional protocol* the subjects make requests one after the other with precise information concerning their position (5 positions) in the sequence order but not about the requests of previous players.
- Under the *cumulative protocol* the subjects make requests one after the other with no exact information about their position (5 positions) in the sequence order but with information about the total amount of previous requests.

In the sequential, positional, and cumulative protocol the harvesting requests were inversely related to the position in the sequence (Budescu, Au, & Chen 1997: 191). The subjects being able to request first harvested the most and the subjects in the last position harvested the least.

---

[338] What is here considered to be sequential decision making is termed by Budescu et al. (1997) as a nonsimultaneous protocol.

[339] A sequential protocol of play is like the cumulative and the positional protocol a special version of a nonsimultaneous protocol.

[340] In games of timing the decision point has to be determined by the players themselves (Dresher 1961: 128-144). In such games each player wishes to delay the decision as long as possible but waiting too long may be penalized. For example in duels each duelist wishes to hold fire as long as possible because his accuracy increases with time. At the same time, if the duelist holds his fire too long his opponent may win the duel.

The above discussed influence of the protocol may be explained by two different lines of reasoning. Sequential decision making results *firstly* in informational asymmetries and *secondly* in commitment asymmetries (Harrison & Hirshleifer 1989: 208). These two very important points will be discussed in turn.

Information asymmetry

There are two aspects why information is changed because of sequential moves:

i. Being second to move eliminates certain outcomes and strategies (Harrison & Hirshleifer 1989: 208). Rational egoists can maximize their welfare by focusing on the best response contingent on the decision of mover one.

ii. Behavior of others influences conditional cooperation of nonegoistic individuals. The individual who moves second knows whether the other individual cooperated or not. In case of a cooperative move the second player may respond reciprocally.

How decision sequence influences information is exemplarily shown with the Stag Hunt game. This game describes a conflict between safety and cooperation in which moving sequentially produces information and in consequence cooperation. The related dilemma was initially described by Jean-Jacques Rousseau (1761).[341] Two individuals go out hunting. For each of them it is possible to go for a stag or a hare. To catch a stag it is necessary that both go for it. At the same time it is possible for a hunter to catch a hare without the help of the other hunter. Though, a hare has less meat than a stag and is therefore less worth even if the stag has to be equally shared. The corresponding payoff matrix for the game played *simultaneously* is the following (see Figure 61):

	Hunter 2 chooses stag	Hunter 2 chooses hare
Hunter 1 chooses stag	4,4	0,3
Hunter 1 chooses hare	3,0	3,3

Figure 61: Payoff matrix Stag Hunt game[342]

With simultaneous moves both stag and hare hunting are Nash equilibriums (Skyrms 2003: 3). Whether to hunt for stags or hares rests on the believe about what the other will choose.

Sequential decision making leads to the replacement of uncertain beliefs by certain knowledge about the behavioral choice of the other hunter (see Figure 62). In this situation rational egoists will go for stags for sure. If Hunter 1 chooses to go for a stag and if Hunter 2 has knowledge about this Hunter 2 will go for a stag

[341] Other examples of coordination problems are people rowing in a boat or neighbors draining a meadow (Hume 1739).

[342] The generic payoff matrix is a > b > = d > c

	Hunter 2 chooses stag	Hunter 2 chooses hare
Hunter 1 chooses stag	a,a	c,b
Hunter 1 chooses hare	b,c	d,d

as well. This reasoning is known to Hunter 1 and he will choose to go for a stag. The conflict between safety and cooperation disappears completely because of an informational change.

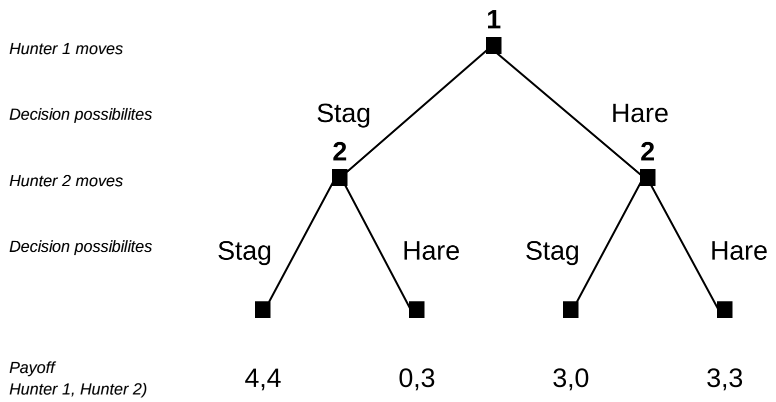


Figure 62: Extensive form of a sequential Stag Hunt game

In the example of the stag hunt game the information advantage of player two is beneficial for both players. This is not necessarily the case. It is possible to construct a game based on different nontransitive dices in which the starting player always looses with the probability of 2/3 if the dices are thrown even he is allowed to choose first from a given and fixed set of dices (Savage 1994: 429). In such a game there are for example three different nontransitive dices from which the starting player may freely chooses one (see also the Condorcet paradox in chapter 5.3). The second player is then always able to choose from the remaining two dices the one that beats player one with the probability of 2/3 irrespective which dice player one has chosen. In this example the information advantage of the second player is not beneficial for player one.

The stag hunt game is an example in which sequentially moves leads to cooperation of rational egoists. It was argued above that for rational egoists the dominant strategy is not altered in games like the prisoner's dilemma. The additional information about the move of player one does *rationally* not make any difference for player two because the chosen strategy does not depend on what the other player does. However, even for this sort of games chosen action may change if there is some knowledge about the move of player one. The prerequisite is that individuals have preferences for fairness or reciprocation (see 5.4.2.1.2 and 5.5.2). If there is such a preference sequential moves will lead to universal cooperation if player one starts cooperatively. Moving first cooperatively gives player two the information that he will not be cheated.

Commitment asymmetry

The second mover has an informational advantage over the first mover but also a severe disadvantage. He has no ability to make a credible commitment to a certain strategy (Harrison & Hirshleifer 1989: 208).[343] Moving first is in fact a very strong commitment because it cannot be reversed. By commitment to a sequence of

343 An illustration of a similar problem in which making a credible commitment is important is provided by Schelling (1960: 43-44). A kidnapper gets cold feet and wants to set his victim free. However, he is afraid that the victim will go to the police. The victim promises not to do so in return for freedom. The problem is now that the victim will have no incentives to keep his promises once he is in freedom. In this situation the victim should disclose a major past offense. This provides an incentive to the hostage to not tell on the kidnapper because he would retaliate and unhide the offense to the police.

actions a player may be able to alter the play of his opponents (Fudenberg & Tirole 1991: 75). This possibility of influencing opponents may illustrated by a general burning the bridge behind him. He thereby commits very credibly not to retreat. In the Spanish conquest of Mexico the commander Hernándo Cortés ordered to scuttle the ship to force his soldiers to fight seriously and to eliminate any means of desertion. Odysseus lashed himself to a mast to exclude his own liberty of action. With these actions the costs of turning back or flight are very high respectively it is not possible doing so.

In the Stag Hunt game the second player had an advantage over player one. He knew about the decision of player one. In this case the information asymmetry was more important than the commitment asymmetry. This is reversed in the big monkey and little monkey game (Gintis 2000: 3-9).[344] In this game both monkeys eat normally fruits from ground level bushes. However, another important part of their nutrition is the warifruit from the waritree. Warifruits have 10 kilocalories of energy. Waritrees produce only sporadically fruit at the very tip of the tree and only one fruit per tree. To get the fruit, at least one monkey must climb the tree. The cost for climbing the tree is 2 kilocalories for a big monkey and almost nothing for a little monkey. The little monkey is able to *descend* much faster from the tree than the big monkey and he is more skillful so that the fruit does not fall to the ground immediately if it is approached. The big monkey is impetuous so that the waritree falls to the ground immediately if he tries to catch it.

If both monkeys climb the tree simultaneously, the little monkey is back on the ground first and is able to eat 3 kilocalories before he is displaced by the stronger big monkey who eats the remaining 7 kilocalories. The net gain for the big monkey is therefore 7 minus 2 kilocalories for climbing. The little monkey gains 3 kilocalories. If only the big monkey climbs the little monkey has more time before he is displaced and eats 4 kilocalories. In the case in which only the little monkey climbs he is able to eat 1 kilocalorie from the fruit while it is still hanging on the tree before it falls to the ground. The big monkey gets 9 kilocalories. Both monkeys have to decide whether to climb or not. The strategy of the big and the little monkey to maximize their net energy rests heavily on the order of decision making.[345] If the big monkey has to make his decision first he knows that if he waits and does not climb the little monkey will climb because he prefers one to zero kilocalories (see Figure 63). This results in the highest possible outcome for the big monkey and that is why the big monkey waits and forces the little monkey to climb.

[344] In the experiments on weak link games and best shot games of Harrison and Hirshleifer (1989) already mentioned in the section on self-efficacy similarities between the best shot game and the big monkey and little monkey game can be found. In the best shot game only somebody must contribute to the public good in order to produce it. Player one can defect and make a credible commitment not to contribute which forces player two to contribute. In the weak link game both players have to contribute in order to get the good produced. In this case player one must contribute as well and there is no commitment advantage.

[345] To learn more about the presented problem it is not important whether monkeys in reality maximize anything, know the costs and benefits of climbing, or are smart enough to find the optimal solution (Gintis 2000: 4).

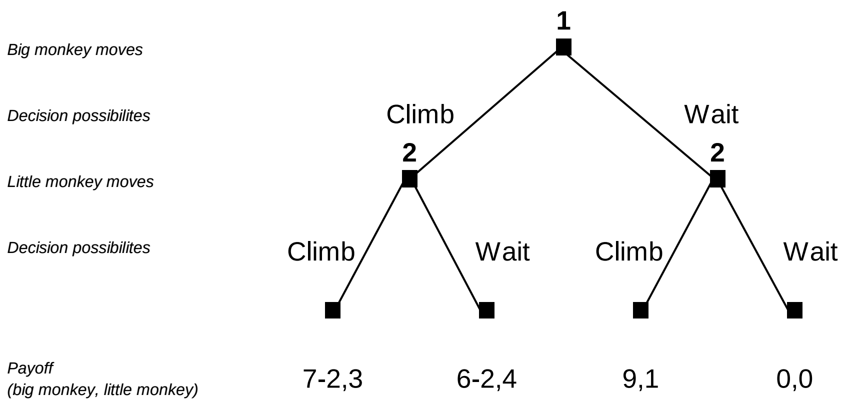


Figure 63: Extensive form of the game if the big monkey moves first

If the little monkey is able to move first he will wait, too. It is more advantageous for the big monkey to climb than to wait which maximizes the net gain of the little monkey (see Figure 64). Therefore, if this game is played sequentially the first mover, irrelevant whether it is the big or the little monkey, is able to force the second mover to choose his most preferred outcome.

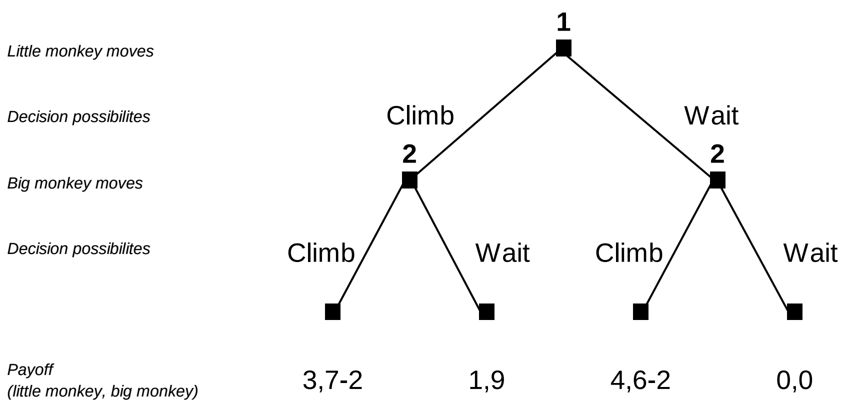


Figure 64: Extensive form of the game if the little monkey moves first

In the case of simultaneous decision making there are three Nash equilibriums (see Figure 65). There is no clear strategy for the monkeys anymore because their best decisions rest on the decision of the other monkey that is not known.

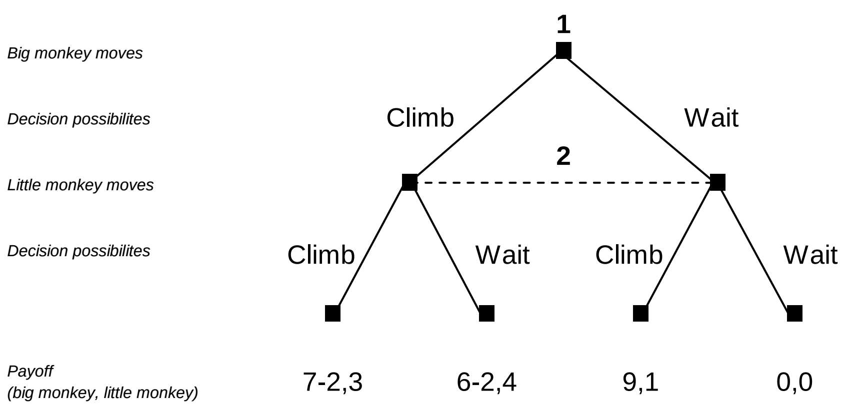


Figure 65: Extensive form of the game if the monkeys move simultaneously

As shown with the stag hunt respectively the big monkey and little monkey game the sequence of decision making may be highly influential for the payoff of the involved individuals. Moving first has the advantage that a credible commitment can be made while moving second has the advantage that there is information about the decision of the other individual.

In a study on collective action Marwell and Oliver (1993) concluded that the most important determinant of collective action are the interest and resource levels relative to the cost of contributing. If contribution is expensive, fewer individuals will do so. If decision making is sequentially individuals may contribute only a small amount of a scare resource and wait whether the others' reciprocate. Thereby information about the others' can be collected. If they do not cooperate not much is lost. Otherwise a chain of cooperation may start producing the collective good entirely. In one-shot games with simultaneous decision making such mechanisms and conditional cooperation are not applicable.

Decision sequence in practice

There are both situations in practice in which decisions have to be made sequentially or simultaneously. However, it seems to be that sequential decision making is more important.

Game theory in its extensive form has been used to analyze the dynamic situations in entry and entry deterrence of industrial organizations (Fudenberg & Tirole 1991: 68). The influence of sequential decision making is displayed in strategic management literature on first mover (dis-)advantages.[346] First movers influence followers by occupying geographical, technological, and customer perceptual space (Lieberman & Montgomery 1998: 1112-1113).[347] Making heavy investments in technology or customers which are lost if a firm withdraw from business is a very powerful *commitment* to stay in business. A firm that invested all its resources in the development of a product has no other possibility than to fight for success that is not a pleasant situation for followers respectively for a prospective competitor. First movers may also mold the cost structure of customers by building up switching costs or establishing network externalities that enables the creation of industry standards.[348] Beside preemption of resources and establishing standards first movers also gain higher experience in organizational and technological capabilities. It may therefore be that followers are restrained in their strategic choice and therefore in their wealth maximization. First movers have a *commitment advantage*.

Yet, the danger for first movers is that because of technological and market uncertainties the wrong resources were acquired. Followers have an *informational advantage* because they can learn from the mistakes of the first mover.

Whether it is advantageous for early market entry rests among other things on strengths and weakness of a firm. Firms with strength in marketing and manufacturing should enter later and firms with strength in new product development should enter earlier. Based on this argumentation the timing of decision making is

[346] Related to the discussion about first mover (dis-)advantage is the Stackelberg game in which a leader and a follower compete on the production output level (Fudenberg & Tirole 1991: 70).

[347] However, it is not completely clear whether market entry is an endogenous (a choice of the firm) or an exogenous variable (Lieberman & Montgomery 1998: 1113).

[348] It is to note that switching costs and industry standards are characteristics that do not reside within the firm but rather at the level of the customers (Lieberman & Montgomery 1998: 1113).

important irrelevant whether it is advantageous to enter early or late. First movers have the advantage that they can make a powerful *commitment* while late movers have *informational advantages*.

Response option

Another aspect influencing behavior in social dilemmas is the available responses. In most dilemma experiments subjects are forced to play with the opponent (Weber, Kopelman, & Messick 2004: 293). This does not represent most real-life situations. Outside of prisons, mental hospitals, schools, ghettos, and the military human beings usually do not have to interact with each other in either case if they do not want (Orbell & Dawes 1993: 787). Human beings spend a lot of energy for maneuvering out of bad games. Certainly, not interacting may be very costly to the individual and some may not afford doing so. However, one may assume that a vast amount of individuals have the opportunity not to interact with an other individual. Interestingly, this option is hardly given to subjects in experiments. Subjects are in the situation of the prisoner's in the prisoner's dilemma with the option to cooperate or to defect but not to refuse playing the game at all (Orbell & Dawes 1993: 788).

Orbell and Dawes (1993: 789-790) highlight the possibility of an exit option with the example of collaborative university research. Faculty collaboration is beneficial to the whole university and therefore to each researcher because individual energy and creativity multiply each other. However, exploiting other faculty members and not contributing time and energy to the collective research is even more beneficial to the individual researcher as long all others contribute. The standard solution for the university management is to reward cooperators. The university might determine the researchers who contributed and allocate money or merit accordingly. The problem is that university administrators have to know a lot about the research field such as the "asymptotic freedom in the theory of the strong interaction" (physics) or the "molecular basis of eukaryotic transcription" (chemistry) to make a good decision. Another solution is to let the faculty members choosing freely among potential collaborators. This solution is based on the assumption that the faculty members know (own experience) or assume (rumors or reputation) which of their fellows are cooperators. If such an exit option is possible one could expect that birds of a feather flock together.

Orbell and Dawes (1993) discuss the question whether giving the choice not playing the game does have an influence on social and individual welfare or not. There are two rationales why cooperators choice more likely to play than defectors (Orbell & Dawes 1993: 788-789):

i. Cooperators choose more likely to play if there is some degree of translucency in which a player's intention to cooperate or defect is recognizable (Orbell & Dawes 1991: 516). Of course, this contradicts the fundamental characteristic in prisoner's dilemma games that cooperative relationships are risky (Orbell & Dawes 1993: 788). However, that emotions are not totally controllable by most individuals seems to be realistic. Social contract theory suggests that "the human mind contains an evolved neurocognitive system that is functionally specialized for reasoning about social exchange, with a subroutine for detecting cheaters" (Cosmides et al. 2005: 505.). Because of natural selection humans have 'algorithms' specialized for thinking about social exchange (Cosmides 1989: 260). Based on several experiments on the Wason selection task[349] the authors of a study conclude that adult subjects are very clever at detecting potential cheaters even if the situation is unfamiliar or culturally

[349] The Wason selection task is about logic.

alien (Cosmides 1989: 262). In consequence, there is some degree of translucency and cooperators may identify each other. Cooperators choice to play with cooperators and refuse to play with defectors. At the same time defectors will not play with each other. In consequence, on the average cooperators choose more often than defectors to play the game if there is an option for doing so.

ii. More cooperators choose to play if subjects' expectations about others depend on their own choice.[350] In this case human beings expect others to be like themselves. In fact, subjects who cooperate expect much higher cooperation rates from others than defectors do (Dawes, Mctavish, & Shaklee 1977). Introspection based pessimism of defectors and introspection based optimism of cooperators increase in consequence the fraction of cooperative subjects if there is an exit option in social dilemma games.[351]

In two different experiments the average payoff of all participants was significantly higher if there was an exit option (Orbell & Dawes 1993: 793). This increase in welfare occurred because cooperators were more willing to enter such games than defectors (Orbell & Dawes 1993: 798). Because of this the frequency of beneficial cooperative-cooperative pairings was higher. Based on the presented results above Orbell and Dawes (1993: 798) argue that the willingness to enter the prisoner's dilemma game implies trust in the partners' cooperative behavior that relies in turn to ones own trustworthiness.

5.4.2.1.2 Social structure

The interaction of individuals hardly occurs in social vacuums (Buckley, Burns, & Meeker 1974: 278). Not only personal characteristics, perceptual factors, or the decision structure influence cooperative behavior but also the social structure. As discussed below communication, group size, group dynamics, fairness, power, leadership, and culture may promote or hinder cooperation.

Communication

That group discussion may enhance cooperation in social dilemmas is well established (Komorita & Parks 1995: 194). During World War I singing Christmas carols together seemed to be a major factor that contributed to mutual trust between the antagonized soldiers and lead to the famous Christmas truce (see chapter 5.5.1.3). As Ostrom et al. (1992: 406) conclude "communication makes a big difference in behavior". Dawes and Thaler (1988: 193) state that "one of the most powerful methods for inducing cooperation in these games {the authors refer to a certain social dilemma game} is to permit the subjects to talk to on another". However, in a one shot social dilemma experiment (Orbell, Vandekragt, & Dawes 1988: 818) not all group discussions had the same positive effect on cooperation. Discussions in which participants unanimously promised to cooperate were especially successful in promoting cooperative behavior. It can therefore be concluded that not only discussion per se but also its content is important for cooperative behavior.

[350] This may be a rational or an irrational bias (Orbell & Dawes 1993: 789).

[351] However, the more likely a defector encounters a cooperator the higher is his average payoff from defection. If nobody cooperates the average payoff of defectors is small in contrast to the situation in that every other player cooperates. The value of defecting is therefore positively correlated with the fraction of cooperative subjects. In consequence, learning techniques to control the own emotions or to overcome introspection may pay off if there are enough cooperators.

There are various and sometimes controversial explanations why communication is effective (Messick & Brewer 1983: 22-29). Communication may enhance group identity (Orbell, Vandekragt, & Dawes 1988), give the possibility for coordination, or initiate norms of cooperation (Kerr 1995: 36-39; Messick & Brewer 1983: 23). However, in many games the substantial improvement in cooperation happens without any possible explanation in terms of the economic model because it does not change the dominant strategy (Orbell, Vandekragt, & Dawes 1988: 811).[352]

A question relevant to this doctoral thesis is whether computer-mediated or face-to-face communication makes a difference in social dilemmas. In social dilemmas it needs trust that the own cooperative decision is not exploited by defecting choices of others.[353] It is usually thought that trust needs touch (Handy 1995). If this were true it would represent a problem for distributed work teams (Wilson, Straus, & McEvily 2006: 16). In distributed work teams there is behavioral invisibility and members cannot directly observe the amount of effort others are expending or overhear what team members say when they are interacting with each other (Wilson, Straus, & McEvily 2006: 16). Based on survey research Burt and Knez (1996: 83) suppose that because of gossip mechanisms trust is amplified if the relationship is observed by third parties. Wilson et al. (2006: 17) report the findings of Wichman (1970) about a prisoner's dilemma. In this experiment the greatest trust occurred among participants in close proximity. Trust was especially strong when the subjects could see each other. Emotions may serve then as a cue for the decisions of others (Orbell & Dawes 1991: 516). Human beings may interpret signs like facial expression or voice tone and infer intentions (Frank 1993: 165). The problem now is that there is reduced social context in electronic communication (Sproull & Kiesler 1986: 1505).

In an experiment Wilson and McEvily (2006) studied trust within teams with face-to-face interaction and within computer-mediated teams. In this experiment subjects had to solve different tasks in three sessions within three weeks. Computer-mediated teams had a lower level of trust at the end of the first session than teams with face-to-face interaction (Wilson, Straus, & McEvily 2006: 27). At the end of the three meetings the level of trust in computer-mediated teams increased to the level of face-to-face teams. The authors of the study analyzed additionally the level of cooperation within the teams as a behavioral outcome of trust.[354] They found the same pattern as for trust: cooperation was initially lower in computer-mediated than in face-to-face teams but with time going on the difference disappears. It can therefore be concluded that distributed teams working electronically develop comparable levels of trust and cooperation like face-to-face teams (Wilson, Straus, & McEvily 2006: 29). However, it takes more time to build up trust.

Adrianson and Hjelmquist (1999) researched whether the means of communication do have an influence on cooperation in social dilemmas. They authors come to the conclusion that there is no influence of the communication channel on cooperation in social dilemmas (Adrianson & Hjelmquist 1999: 188).

[352] If there is no anonymity this is only true if the subjects do have no connections with each other besides the ones in the experiment.

[353] This statement is not true for altruists (McClintock 1972) who maximize others outcome. By being exploited this goal is achieved.

[354] Cooperation was measured by the willingness to devote tokens to a team managed stock portfolio instead of spending the money to the individual stock portfolio (Wilson, Straus, & McEvily 2006: 21). There were incentives for both alternatives and decisions were anonymous.

Fairness

According to Rabin (1993: 1281) “people like to help those who are helping them, and to hurt those who are hurting them”.[355/356] People respond usually to social situations they consider to be fair positively and vice versa (Van Prooijen, van den Bos, & Wilke 2004: 33). Negative responses may include revenge (Bies & Tripp 2001), legal action (Lind et al. 2000), stealing (Greenberg 1997), or aggressive behavior (Folger & Skarlicki 1998). The result of such behavior may be Pareto damaging such as the rejection of unfair offers in ultimatum games. The influence of fairness on groups is, as Cropanzano et al. (2001: 183) conclude, that “justice pulls us together, and injustice pushes us apart”. In fact, Charness and Rabin (2002: 847) conclude that “an analysis over a broad range of games indicates that reciprocity considerations are an important component of behavior”. Cooperative behavior can in this case not be attributed to pure altruism where individuals cooperate *unconditionally* (Rabin 1993: 1283). The willingness being cooperative is in such cases *contingent* on behavior of other individuals.[357] However, van der Heijden et al. (1998: 31) found few signs for reciprocity in their gift giving experiments.

De Vries and Wilke (1992: 83) propose that fairness considerations are crucial in social dilemmas.[358] Individuals therefore try to maximize their own payoff *provided* that this is done fairly. This means that individuals do not exploit others to enhance their own benefit. De Vries and Wilke argue that fairness does not refer to a fair distribution of payoffs but to a restriction on own gain maximization. Eek and Biel (2003) discuss the influence of fairness in collective good dilemmas with the “GEF” hypothesis. They assume that behavior is driven by the three motives **g**reed, **e**fficiency, and **f**airness. Following greed individuals try to maximize their payoff and leave very little for others. However, greed is constrained by considerations of efficiency and fairness. The former constraint refers to the problem that resources may be deteriorated if subjects are too greedy. The later constraint refers to the motivation not to receive more than others do. Both efficiency and fairness constrain greedy harvesting. The authors conducted two experiments to study the interplay between greed, efficiency, and fairness and its effect on cooperation. They concluded that “when people have to choose between cooperation and defection, they are to a large extent affected by fairness considerations” (Eek & Biel 2003: 211).

There are several theories why individuals care about fairness (Cropanzano et al. 2001). In the relational model the inclusion within a group provides self-worth and identity (Cropanzano et al. 2001: 174). Being a valued and respected member of a group is a basic need (Van Prooijen, van den Bos, & Wilke 2004: 36). However, individuals need information about their status in the group. They can find such information by looking how they are treated by group members. Fair treatment conveys information about the relationship to the group (Cropanzano et al. 2001: 174).

---

355 Fairness seems to be not only relevant for human beings but also for monkeys. In an experiment on food giving among genetically unrelated cotton-top tamarins the authors showed that the monkeys donate food even though they obtain no immediate benefit from doing so (Hauser et al. 2003: 2368). However, the extent to which this behavior is shown is contingent on other’s donating behavior. The monkeys did thereby distinguish between altruistic food giving as opposed to giving as a by-product of selfishness (Hauser et al. 2003: 2363). This result suggests that the donation of the monkeys depends upon the perceived motivations of other monkeys (Hauser et al. 2003: 2366).

356 In an experiment of Dawes et al. (2007) researched the question whether the willingness to alter other’s income in social dilemmas by punishment or reward is motivated by the aim to promote cooperation or the aim to produce equality. They found that that there are indeed egalitarian motives that caused punishment or reward.

357 Cooperative behavior may also be contingent on the neediness of others (see social motives in chapter 5.4.1).

358 Interestingly, Knorr Cetina and Bruegger (2002: 924-928) found that reciprocity is also important in interbank currency trading of global investment banks.

There are different branches of fairness (Cropanzano & Byrne 2000: 143-144; Cropanzano et al. 2001: 165; Gilliland & Chan 2001; Van den Bos 1999: 561; Van Prooijen, van den Bos, & Wilke 2004: 34):

- One branch refers to fair formal processes and ***procedural justice*** (Thibaut & Walker 1975).[359] Procedural fairness is related to the processes leading to an outcome. A finding of a series of experiments of de Cremer et al. (2005: 399) is that procedural fairness is an important tool to manage cooperation within groups. The effects of procedural fairness are robust across methodologies and samples (Van Prooijen, van den Bos, & Wilke 2004: 35).

  However, there are different and divergent interpretations what fair processes are (Leventhal 1980: 104-105; Thibaut & Walker 1975; Tyler 1988). Thibaut and Walker (1975) focused on control procedures. Folger (1977) shifted the focus from process control to the opportunity to voice the own opinion. Decision making processes in which there is the opportunity to voice the opinion are considered fairer than processes that do not (Van Prooijen, van den Bos, & Wilke 2004: 35). The opportunity to voice the own opinion in decision making is the most generally accepted factor in the procedural justice domain (Van den Bos 1999: 561).

  In another approach of fair processes Leventhal (1980) identified six criteria for fairness specially adapted to organizational processes (Greenberg 1990: 401):

  i. Consistency that refers to similarity of treatment and outcome across time and people.
  ii. The ability to suppress bias that prevents favoritism and the like.
  iii. Decision quality or accuracy.
  iv. Improvability respectively the opportunity to correct unfair or inaccurate decisions.
  v. Representation refers to involvement in decision making.
  vi. Ethicality that rests on general standards of fairness and morality.

  In reality, the definition of fair processes varies according to the nature of the situation (Tyler 1988: 104). Consistency, accuracy, and bias suppression are for example considered more significant in formal than in informal settings (Barretthoward & Tyler 1986). Tyler (1988: 131) raises in consequence the question whether there are universally fair procedures at all.

  Barretthoward and Tyler (1986) argue that procedural justice is most important in unstable but desirable relationships and the least important in interpersonal relationships not threatened by injustice or relationships in which there are strong positive bonds.

- Another branch refers to fair outcomes and ***distributive justice*** (Adams 1963). Distributive fairness is related to the outcome allocated to people. The outcome of a court trial and the manner in which the trial is conducted are two separable aspects of fairness (Van den Bos 1999: 561). Thibaut and Walker (1975) showed that people's justice concern refer to procedural as well as distributive justice.

  There are different rules for fair distribution (Colella 2001):

  i. The *equality* rule states that everyone should be treated equally respectively that everyone should get exactly the same share (Deutsch 1975).

[359] For reflections on the relationship between legitimacy and procedures see Luhmann (1969).

ii. The *equity* rule states that people will evaluate an outcome as fair or unfair by comparing their own input/outcome ratio to the one of others (Adams 1963). In an experiment Chan et al. (1997) showed that fairness respectively equity rules notably influences the contributions to public goods.

iii. The *need* rule urges that fairness is determined by the individual need of a person (Deutsch 1975).

Biel et al. (1997) surveyed 1840 parents' willingness to contribute to municipal childcare resources. Subjects were also asked whether they consider the *equality* rule (the same care for all children), the *equity* rule (childcare according to parent's contribution), or the *need* rule (childcare based on personal need) the fairest. The parents considered an *equality* distribution of child care the most and an *equity* distribution the least fair.

Colella (2001) concludes that the equity rule is the most common type of distributive fairness in work situations. Conversely, according to Stouten et al. (2006: 894) it was shown that in symmetric social dilemmas subjects have preferences for the equality rule.

- Yet another branch of fairness is ***interactional justice*** that refers to decent interpersonal behavior (Bies & Moag 1986; Cropanzano et al. 2001: 165). Interactional justice has been operationalized with i) clear and adequate explanations and ii) treatment with dignity and respect (Cropanzano et al. 2001: 180). The foundation of interactional justice is the observation that fairness not only refers to formal procedures but also to the quality of treatment (Greenberg 1990: 411). It is however questioned whether interactional justice is not a component of procedural justice and some researchers have returned to the basic distinction between fairness of outcome and fairness of processes (Gilliland & Chan 2001: 144-145; Van Prooijen, van den Bos, & Wilke 2004: 181-183). Lind and Tyler (1988) claim that procedural justice includes formal structural as well as interpersonal aspects.

An important factor influencing distributive as well as procedural justice is standing (Van Prooijen, van den Bos, & Wilke 2004: 34). Standing refers to the position or status an individual occupies within a group. An individual may then perceive differences in outcome, e.g. salary, as fair because of his position. Chen et al. (2003: 17) showed that status moderate the interactive relationship between distributive and procedural justice. After conducting a series of experiments van Prooijen et al. (2004: 52) conclude that standing causally influences procedural fairness effects.

Power / leadership

Leadership may be defined as the process wherein a member of a group or an organization influences the interpretation of events, the choice of objectives and strategies, the organization of work activities, and the like (Yukl 1998: 5). There are two reasons why leadership may be favorable for groups (Buckley, Burns, & Meeker 1974: 289): i) a leader might provide selective incentives[360] that alters the payoff associated with a certain action and ii) he may influence individuals' perceptions and evaluations. The former refers to a structural/instrumental and the later to a motivational/relational solution to the problem of free riding in social dilemmas (see chapter 5.5).

[360] To distribute selective incentives the corresponding higher order or meta- prisoner's dilemma must be solved which is discussed in chapter 5.5.1 (Buckley, Burns, & Meeker 1974: 293).

Choosing a leader is not without its own problems. One problem is that doing so is in fact a special form of collective action itself (Van Vugt & de Cremer 1999: 587). Having a good leader is clearly beneficial to everybody but individually it is rational to let others do the investment to find and build up a candidate and organize the election.

Leadership as a structural/instrumental solution

Hobbes' (1998: 114) solution to a social dilemma is to install a Leviathan.[361/362] The Leviathan is the gigantic biblical sea monster that cannot be beaten *by a human being*. It is an allegory for an all-powerful government. This solution overcomes the problem of social dilemmas by hiring an agent. The duty of this agent is to punish anyone breaking rules and therefore increasing costs for defection.

In the state of nature, according to Hobbes, human beings decided freely to put themselves under the tyranny of the Leviathan. It may be therefore instrumental to install a leader and move to a hierarchical group structure if otherwise the group does not succeed (Samuelson & Messick 1986: 961). Individuals give up their freedom to a leader to secure a collective good (Van Vugt & de Cremer 1999: 588). This solution eliminates the dilemma because subjects have either not the choice to defect at all or defection is not favorable (Van Vugt & de Cremer 1999: 587). Groups may therefore create a structure that enforces contribution to a collective good (Van Vugt & de Cremer 1999: 587). Human beings may change interaction situations into ones that are suitable to solve social dilemmas (Buckley, Burns, & Meeker 1974: 278). This is also the proposed solution of Hardin (1968: 1247-1248) who discusses the necessity to create some sort of mutually agreed coercion as a social arrangement to avoid the tragedy of the commons.

Leadership as a motivational/relational solution

Leaders are not only beneficial because of their power to punish and sanction group members. It also may be that leaders influence others to cooperate by their example and their ability to convince followers that the proposed course of action is right and just. Such a leader motivates others to cooperate. If a leader wants to influence subjects to cooperate the attributes of the leader have to correspond with those of the group or the situation (Van Vugt & de Cremer 1999: 588). Leaders with autocratic power who have the freedom to do whatever the person considers appropriate are not very often found in reality. Usually groups consign democratic leaders who can be replaced if it is necessary. This is in sharp contrast to the tyrannical Leviathan.

There are two fundamental ways of consigning leaders. They can either be appointed by some sort of authority or elected by group members. In either case the leader's legitimacy is crucial for success.[363] Tyler and Degoey (1995) showed that legitimacy has an effect on cooperation in social dilemmas. During the 1991 California water shortage people's willingness to restrain themselves was not influenced by the perceived severity but by positive bonds to the authorities and the perceived legitimacy of authorities. It seems that legitimacy enhances

---

[361] According to Hobbes (1998: 114) "every man should say to every man, 'I authorize and give up my right of governing myself, to this man, or to this assembly of men, on this condition, that thou give up thy right to him, and authorize all his actions in like manner'".

[362] Aumann (2005: 357) notes that this solution is also suggested by Isaiah (2:2-4). According to Aumann Isaiah argues that "nations can {only} beat their swords into ploughshares when there is a central government recognized … by all".

[363] In the early nineties conflicts in the BSD community resulted in a split into three competing projects (Raymond 2004: 42). The history of the BSD family shows the importance of an accepted leader and legitimacy.

the ability of a leader to secure compliance with his directives (Tyler 1990). While the severity of resource scarcity explains the willingness to empower a leader, legitimacy explains why the people follow the instructions (Tyler & Degoey 1995: 483).

Alternatives to leadership

In social dilemmas group members prefer a democratic to an autocratic leader and an elected over an appointed leader (Van Vugt & de Cremer 1999: 595). However, subjects demand leadership only if it is necessary (Van Vugt & de Cremer 1999: 595). Samuelson and Messick (1986: 961) report that 60% - 70% of subjects who witnessed ineffective use of a common good voted to elect a leader to manage it. However, subjects who experienced an effective use of the common good voted far less for installing a leader. From these results one may conclude that installing a leader is not the most preferred solution but that it is accepted as a petty evil to evade inefficiency.

Ostrom (1997) suggested installing powerful institutions instead of leadership. She observed that long enduring common pool resources have operational rules that are monitored and sanctioned by the appropriators themselves (Ostrom 1997: 90-100). These operational rules forbid, permit, or require some actions and outcome (Ostrom 1997: 51). Interactions between individuals are established and maintained by the involved individuals themselves (Buckley, Burns, & Meeker 1974: 294). This is a form of self-regulation that must be distinguished to exogenous control.

Group size

In his seminal book Olson (1995: 2) proposes that the problem of free riding in public good dilemmas is related to group size:

> *… unless the number of individuals in a group is quite* ***small*** *{emphasis added}, or unless there is coercion or some other special device to make individuals act in their common interest, rational, self-interested individuals will not act to achieve their common or group interests.*
>
> *The larger the group is, the farther it will fall short of providing an optimal supply of any collective good, and the less likely that it will act to obtain even a minimal amount of such a good. In short, the larger the group, the less it will further its common interests.*

Olson (1995: 21) demands therefore that what it „… needs to be known, in the words of the German sociologist Georg Simmel, is the 'bearing which the number of sociated individuals has upon the form of social life'".[364] Hardin (1982: 38) condenses Olson's argument with the simple assumption: „large groups will fail; small groups may succeed".

For Olson (1995: 53-65) there are three reasons that may explain the suggested relationship between group size and cooperation:

i. The individual net benefit for contributing decreases with group size.

[364] Simmel (1964: 87) states further that "a group upon reaching a certain size must develop forms and organs which serve its maintenance and promotion, but which a smaller group does not need".

ii. Big groups are less likely privileged.[365]

iii. The costs for coordination increases with group size.

Hardin (1982: 38-49) as well as Udehn (1993: 241-242) challenge this reasoning:

i. It is questionable whether the individual net benefit really decreases always with group size. If it were true, the benefits for contributing would have to decrease with group size or the individual costs would have to rise. Although individual benefits may decrease with group size, this is not necessarily the case. In fact public goods are typically characterized by some jointness of supply or non-rivalry (Samuelson 1954: 387).[366] This means that the benefit of a good to an individual is not diminished because another individual use it as well. An example is the lighthouse that serves everybody independent on how many others have benefited from the light. Contrary to the reasoning of Olson individual benefit increases with group size for many public goods. Examples are riots and revolutions (see chapter 6.4). Additionally, the total costs associated for providing a public good must not necessarily rise with group size. For a lighthouse the total costs stay the same. In this case, individual costs decrease with higher group sizes if everybody burdens his fair share.

ii. According to Udehn (1993: 241) there is not automatically a relationship between group size and the probability of being a privileged group. A large group could be as well privileged.

iii. Variable costs for coordination increases with group size. However, important aspects of coordination costs are fixed overhead costs that do not vary with the size of the group. It is more expensive to organize 2000 than 1000 individuals but the costs are probably not doubled. Although coordination costs increase with group size this effect is not always substantial. Setting up a mailing list for coordination for 1000 or 2000 individuals does hardly differ.

Despite the theoretical flaws in Olson's (1995) argumentation many studies corroborated the proposed relationship between group size and cooperation (Bonacich 1976; Fox & Guyer 1977; Hamburger, Guyer, & Fox 1975; Komorita & Lapworth 1982). Messick and McClelland (1983) found that groups of 3 subjects performed better than groups of 6 subjects. Hamburger et al. (1975) found that cooperation in groups with 3 persons was higher than in groups of 7 persons. Komorita and Lapworth (1982) discovered that for groups with 2, 3, and 6 person cooperation decreases with increasing group size. Brewer and Kramer (1986: 543) found no effect of group size on cooperation in common good dilemmas. In public good dilemmas cooperation was lower in big groups.

However, the relationship between group size and cooperation seems to be more complicated than suggested by Olson: Group size influences cooperation until a certain point negatively (Fox & Guyer 1977; Komorita & Lapworth 1982; Liebrand 1984; Messick & Mcclelland 1983). After that point increasing group size does not have any influence on cooperation anymore. Liebrand (1984) found no differences in cooperation in a social dilemma between groups consisting of 7 respectively 20 individuals.

Kerr (1989) suggested that empirical results about decreasing cooperation in large groups could be explained by self-efficacy (see also chapter 5.4.2.1.1). This is the belief, irrespective whether true or not, that it is in one owns capacity to achieve a particular outcome. Even if group size had no objective impact on the individual

[365] According to Olson (1995: 50) in a privileged group "each of its members, or at least some of them, has an incentive to see that collective good is provided, even if he has to bear the full burden of providing himself".

[366] There is no jointness of supply in common good dilemmas.

ability to achieve a goal individuals in small groups feel more capable to make a difference. Concerning group size individuals have an illusion about efficacy and assume that free riding increases with group size.

Culture

We know from various research that the culture of nations differ (Bond et al. 2004; Hofstede 1980; House & Global Leadership and Organizational Behavior Effectiveness Research Program. 2004; Schwartz 1994; Welzel, Inglehart, & Klingemann 2003). Culture is relevant for business because it influences organizational live, work-related values (Fletcher & Perry 2001: 131), job satisfaction (Judge et al. 2001)[367], corporate governance (Dore 2005), and extra-role behavior (Fischer et al. 2005).[368] This has consequences for the management. Elenkov (1998) argues that a management concept cannot be applied unchanged to different cultures.

An especially influential categorization of culture has been developed by Hofstede (1980; 2001). Based on a study in 40 countries with a total of 116’000 questionnaires he identified four main dimensions by which the value system of a culture can be described. As a reaction of the Chinese Value Survey Hofstede (2001: 351) added later a fifth dimension. These five dimensions are:

- ***Power distance*** is the degree to which less powerful individuals expect and accept that the power is distributed unequally. While Malaysia scores high on power distance and power inequality is therefore accepted, Sweden scores low.
- ***Individualism and collectivism*** describes the extent to which individuals are integrated in groups. The United States is a very individualistic society stressing personal achievements, individual rights, and individual interests. Guatemala on the contrary is a very collectivistic society stressing group ties and group interests.
- ***Uncertainty avoidance*** deals with the tolerance of uncertainty and ambiguity respectively whether individual fell comfortable in unstructured and surprising situations. Japan scores high on uncertainty avoidance. By using law, rules, and philosophical believes uncertainty is avoided. In contrast Singapore scores low on uncertainty avoidance indicating that its citizen fell comfortable in uncertain situations.
- ***Masculinity and femininity*** refers to roles usually attributed to gender. A masculine culture such as Italy stress assertiveness, power, strength, and the like while feminine cultures such as the Netherlands stress the value of relationships and quality of life.
- ***Short- and long-term orientation*** describes the importance attached to the past and the present versus the importance of the future. In long-term oriented cultures thrift and perseverance is valued while in short-term oriented cultures the respect for tradition and reciprocation of gifts is more dominant. While China is long-term oriented Sierra Leone is short-term oriented.[369]

[367] Interestingly, samples across nations indicate that individualism has a positive link to job satisfaction while studies within a country found that collectivism correlates positively to job satisfaction (Judge et al. 2001: 38).

[368] However, Gelade et al. (2006: 552) found no significant relationship between organization commitment and collectivism.

[369] For short- and long-term orientation only 23 countries were surveyed.

That culture also influences behavior in social dilemmas is exemplified by an interesting experiment on self-interest conducted in 15 small societies in South America, Africa, and Asia (Henrich et al. 2005). It was researched whether there is a cross-cultural difference in selfish behavior. The same ultimatum, public good, and dictator games were played in very different cultural settings. The authors summarized their findings as follows (Henrich et al. 2005: 797-798):

i. There is no society in which experimental behavior is fully consistent with the selfishness axiom.

ii. There is much more variation between groups than observed in other studies.

iii. Differences between societies in market integration (frequency in market exchange) and the local importance of cooperation explain a substantial portion of the behavioral variation between groups.

iv. Economic and demographic variables on the individual level do not consistently explain behavior within or across groups.

v. Behavior in experiments often reflects patterns of interaction found in everyday life.

Some of these findings are based on an ultimatum game that is briefly discussed. In the ultimatum game subjects are anonymously paired and the first player, called the proposer, is provisionally given a substantial amount of money. The proposer must offer a portion of this amount to the second player who is called responder. The responder can either accept or reject this offer. In the first case the responders keeps the offer and the proposer the portion not offered to the responder. In the second case nobody gets anything.

Group	Number of Pairs	Mean offers	Percent of offers rejected
Lamalera	19	57%	n/a[a]
Aché	51	48%	0%
Shona (resettled)	86	45%	7%
Shona (all)	117	44%	8%
Orma	56	44%	4%
Au	30	43%	27%
Achuar	14	43%	n/a[a]
Sangu (herders)	20	42%	5%
Sangu (farmers)	20	41%	25%
Sangu	40	41%	15%
Shona (unresettled)	31	41%	10%
Hadza (big camp)	26	40%	19%
Gnau	25	38%	40%
Tsimane	70	37%	0%
Kazakh	10	36%	0%
Torguud	10	35%	10%
Mapuche	31	34%	6%
Hadza (all camp)	55	33%	24%
Hadza (small camp)	29	27%	28%
Quichua	15	25%	n/a[a]
Machiguenga	21	26%	5%

[a] The rejection rates in these societies are not comparable to the ones of the other societies because of a variation in the game setting.

Figure 66: Results of ultimatum game experiments[370]

[370] Henrich et al. (2005: 802)

The proposers in the conducted ultimatum game do not make offers consistent with standard game theoretical predictions based on the selfishness axiom. In that case proposers should offer the smallest positive amount because if responders maximize their income they will accept every positive amount. As Figure 66 shows the mean offers did not go below 25% of the total amount of money. There are also only for the Aché, the Tsimane, and the Kazakh zero rejection rates of offers.

The offers both of the Achés and of Aus are quite high on the average. While in the former society nobody rejects these high offers, in the later society the rejection rate is quite high. The same pattern holds true if the initial offer is small. In this situation the Hazdas reject very often in contrast to the Machiguengas.

Behavior not in line with selfish maximization of income like the portion of money offered by the proposer in the ultimatum game can be partly explained with two factors. There is higher cooperation in cultural settings in which subjects are i) customized working together and ii) dependant on cooperation. In consequence, it can be hypothesized that historical trajectories of cooperation influences cooperation in the future.

Yamagishi (1988a; Yamagishi, Cook, & Watabe 1998) developed an interesting distinction between an ***individualistic*** and an ***institutional view of culture***:

- In the ***individualistic view of culture*** individuals have a preference for collectivism or individualism. In this case individuals from collectivistic countries such as Japan free ride to a lesser extent because it hurts the nation, the community, the group, or the individual. Actually, cooperation in collectivistic countries refers to social motives already discussed. Successful cooperation is then based on motivational solutions such as building up social norms (see chapter 5.5.2).[371]
- In the ***institutional view of culture*** a collectivistic solution prevents free riding. More precisely, there is a tight system of mutual monitoring and sanctioning that prevents free riding. Consequently, this view assumes that cooperation of individuals form collectivistic countries will collapse if there are no appropriate institutions. The decline of cooperation after institutions are removed will be especially high for individuals from collectivistic countries because strong sanctioning i) undermines intrinsic motivation of cooperation and ii) makes individuals less trustful of others. Cooperation in collectivistic countries refers to payoffs shaped by sanctions as already discussed. Successful cooperation is then based on structural solutions such as negative selective incentives (see chapter 5.5.1).

Benedict (1989) argues that in the Japanese culture behavior is controlled externally by sanctions while it in the United States it is controlled internally by guilt and shame. This may explain why the average level of general trust of Japanese individuals is lower than the one of individuals from the United States (Yamagishi 1988b; Yamagishi & Yamagishi 1994). American subjects cooperated more in a social dilemma than Japanese subjects if there are no sanctions (Yamagishi 1986). In contrast to American subjects Japanese subjects also opted more often for exit from a social dilemma situation respectively from collective rewarding in order to be rewarded individually (Yamagishi 1988a). Similarly, Conway et al. (2006) found that collectivistic societies tend more to initiate political and legal restrictions to the individual. Therefore, the institutional view of culture seems to be more influential shaping cooperation than the individualistic view.

[371] Yamagishi (1988b: 265) calls cooperation based on sanctioning *instrumental cooperation* and cooperation without sanctioning *elementary cooperation.*

Group dynamics

Weber et al. (2004: 298) suggest that groups add an important dynamic to social dilemma settings. Group identification is important as an experiment of Kramer and Brewer showed (1986). Indications that a common resources was depleted by a member of the own group prompted subjects to harvest less. However, if the common resources were depleted by a member of another group subjects started to harvest more.

In a field experiment high school students were asked to pick oranges in groups of four subjects (Erev, Bornstein, & Galili 1993). There were three different payoff conditions namely personal reward, collective reward, and an intergroup competition with a reward to the most efficient team. If the subjects were rewarded collectively it resulted on average in a 30% production loss compared to the personal payoff rule. This production loss was eliminated if intergroup competition was introduced. This effect was the more salient the more similar the competing teams were with respect to overall abilities. These findings are in line with a huge amount of literature on the interindividual - intergroup discontinuity effect. A meta-analysis about this effect showed that intergroup interactions in social dilemmas are generally more competitive or noncooperative than interindividual interactions (Wildschut et al. 2003: 698). This may be explained with the assumption that groups offer social support of individual's disposition of selfishness (Insko et al. 1990: 69). It also may be that individuals with high influence in the group are the ones who favor competition. Dawes and Messick (2000: 114) noted that in social dilemmas the tendency to support one's own group can result in negative outcomes in the end.[372]

5.4.2.2 Perceptual factors

Granovetter (1985) criticized that theories on collective action focus solely on rational cost-benefit analysis to the exclusion of social and emotional factors. Not only the structure but also the context of a situation is important (Poppe 2005). Individuals with similar antecedents are affected differently by similar events (Marshall 1910: 18). As Hayek (1979: 80) observes one should not "take it for granted that what to us appears alike or different will also appear so to others". This is hardly ever taken into consideration in economic experiments. In these experiments payoffs are considered relevant and not how persons interpret the experiment or an event.

Context

Several experiments corroborated that even if the dilemma structure is the same different levels of contribution are observed in different contexts. For example in a prisoner's dilemma game participants were divided into two groups (Liberman, Samuels, & Ross 2004). The game played shared in both groups the same structure, instruction, and payoff matrix. However, the context of the played game differed insofar as the label of the game was distinct. Participants were told either that they play a Community game or that they play a Wall Street game.[373] Although only the name of the game was labeled differently the cooperation rate dropped

[372] That is probably why the fourth President of the United States James Madison wrote under the pseudonym Publius (Hamilton et al. 2005: 301) in the Federalist paper 55: "In all very numerous assemblies, of whatever character composed, passion never fails to wrest the scepter from reason."

[373] In both situations no further information regarding the title was given.

from 66% in the Community game to 32% in the Wall Street game (Liberman, Samuels, & Ross 2004: 1177).[374] The authors replicated the results with Israeli pilot trainees (Liberman, Samuels, & Ross 2004: 1178-1180). The average cooperation rate dropped from 45% in the community situation to 25% in the Wall-Street situation. Interestingly, air force instructors who were asked to predict the behaviour of their students were not able to anticipate this effect.

Another experiment on social dilemmas explored not only the influence of context on cooperation but also its interaction with feedback (Pillutla & Chen 1999). Chen and Pillutla (1999) created an economic and a non-economic context. Participants were asked to take on the role of a factory representative and contribute to a joint investment fund (economic context) respectively to a joint social event (noneconomic context). While both tasks were embedded within a business setting the decision context varied. The payoff matrices of the two scenarios were equivalent. Participants were randomly divided into six groups. Group A, B, and C played in round one the scenario with the economic context while group D, E, and F the one with the noneconomic context. After their contribution in round one participants received competitive feedback (other group participants invested nothing; group A and D), cooperative feedback (other group participants invested the maximum, group B and E), or no feedback (group C and F). In round two participants played the other scenario (see Figure 67).

Conditions	Time 1		Feedback		Time 2
A	Economic context (a1)	▶	Competitive	▶	Noneconomic context (a2)
B	Economic context (b1)	▶	Cooperative	▶	Noneconomic context (b2)
C	Economic context (c1)	▶	None	▶	Noneconomic context (c2)
D	Noneconomic context (d1)	▶	Cooperative	▶	Economic context (d2)
E	Noneconomic context (e1)	▶	Competitive	▶	Economic context (e2
F	Noneconomic context (f1)	▶	None	▶	Economic context (f2)

Figure 67: The experimental design of Pillutla and Chen[375]

Pillutla and Chen (1999: 90) hypothesize that the context in which a social dilemma is embedded is important because it provokes a different norm. Self-interested behavior as an implicit norm is more important in an economic context than in a noneconomic context. The authors suggest further that when a norm inferred from the situation is consistent with the observed behavior people's reaction is less dramatic than if there is an inconsistency between inferred norm and observed behavior. In a noneconomic context cooperative and in an economic context competitive behavior is expected. Participants should then be positively surprised about

[374] The result should urge scientists to pay attention whenever inferring from experimental results to real world behaviour. Small differences in the context of Liberman et al.'s (2004) experiment has lead to huge differences in outcome. In experiments concessions about external validity are made to increase internal validity that requires abstraction and simplification to make research tractable (Schram 2005: 226). It may be sometimes viable to accept low external validity. However, if so one has to be "cautious in predicting real life behaviour based on experimental evidence" (Benz & Meier 2005: 11).

[375] Pillutla and Chen (1999: 90).

cooperative feedback in an economic situation and negatively surprised about competitive feedback in a noneconomic situation and adjust their future behavior correspondingly. Because of this line of reasoning the authors hypothesize the following relationships:

i. In round one the mean contribution in conditions A, B, and C are smaller than the one in condition D, E, and F ($a1 + b1 + c1 < d1 + e1 + f1$).

ii. In round two the mean contribution in the noneconomic context following cooperative feedback about the economic context is larger than if feedback was competitive or no feedback was given ($b2 > c2$ and $a2 = c2$).

iii. In round two the mean contribution in the economic context following competitive feedback about the noneconomic context is smaller than if feedback was cooperative or no feedback was given ($e2 < f2$ and $d2 = f2$).

Results of the experiment are (see Figure 68):

i. Cooperation rates were significantly higher in the noneconomic (on average 39.08%) than in the economic context (on average 32.12%).

ii. Cooperation rates increased non-significantly after cooperative feedback about the economic situation.

iii. Cooperation rates dropped dramatically and significantly in the economic context after competitive feedback on the noneconomic situation

Conditions	Context	Time 1		Feedback		Context	Time 2
A	Economic context	32.25	►	Competitive	►	Noneconomic context	29.56
B		29.68	►	Cooperative	►		49.39
C		27.54	►	None	►		36.52
D	Noneconomic context	43.04	►	Cooperative	►	Economic context	51.67
E		39.61	►	Competitive	►		11.69
F		32.08	►	None	►		39.58

Figure 68: Mean percentage of contribution across context[376]

These findings corroborate the assumption that framing and context influences decision making. A further result is that reaction to the violation of social norms may be quite harsh, especially in a noneconomic context. Pillutla and Chen (1999: 99) argue that a solution to the cooperation problem in social dilemmas is to highlight the noneconomic aspects of the situation. However, Pillutla and Chen recall that cooperation in noneconomic contexts is fragile and reactions to a violation of social norms may become very negative.

By variations of presentation, confounding variables, and context researchers could change preferences for certain or uncertain outcome for a decision making problem dramatically (Wagenaar, Keren, & Lichtenstein

[376] Pillutla and Chen (1999: 96).

1988: 186). It was for example asked do decide what to do on an island on which one person of 100 islanders is known to carry a disease (Wagenaar, Gideon, & Lichtenstein 1997; Wagenaar, Keren, & Lichtenstein 1988). Unfortunately, nobody and not even the sick person know who carries the disease. Doing nothing was one possibility with the consequence that everybody has an equal chance of 1% of dying from the disease. Another possibility was to vaccinate everybody with the result that nobody dies due to the disease but somebody will die because of the adverse effects of the vaccination with 100%.[377] The researchers started to vary the cover story without change the probabilities or the decision making structure. In one of the six versions the subjects received in addition to the story a pictorial display to illustrate the choices. In the vaccination condition the display showed 99 upright black stylized figures and one horizontal (dead) figure in red. The do-nothing condition was illustrated by 100 black figures with red dots. Depending on the presented version the approval rate for vaccination differed widely. In a next step the researchers varied the game insofar as the subjects were told that doing nothing means that the infected person dies with 100%. Vaccinate the islander means that everybody has the chance of 1% to die. In yet another step the story was changed to school kids who were taken as a hostage. Without discussing the results in detail it is interesting to note that due to the presented variation the preference of 83% for a *certain* outcome could be changed to a 92% preference for an *uncertain* outcome (Wagenaar, Gideon, & Lichtenstein 1997: 562-563).[378]

To test color vision at night will lead to errors in performance because eyes are simply designed for daytime activities (Wang 1996: 58). Wang argues that testing people's decision making in an irrelevant context leads similarly to misleading results. Research an irrelevant context will lead to results from which no deduction for a practical situation can be made. Social dilemmas should therefore be based on proper context. To introduce relevant context in social dilemma research is, however, neither trivial nor without practical problems. It is questionable whether arms race or crime are relevant scenarios for first year undergraduate students. Subjects familiar with such context are, however, hardly available for researching corresponding social dilemmas.

That it is important for cooperation to create relevant conditions is highlighted by the research on altruism in apes. It was thought for a long time that altruism is unique to humans (Warneken et al. 2007: 1414). However, an interesting series of experiments suggested otherwise (Warneken et al. 2007). In a commentary on the results of Warneken et al. (2007) de Waal (2007: 1407) concludes that all previous experiments have not shown that there is no altruism in apes but "that humans can create situations in which apes focus on their own interests".

Causes

In common good dilemmas it makes according to Samuelson (1991) a difference what subjects think is the cause for the depletion of a resource. If people think that a water shortage is caused by individuals who have harvested inappropriately they increase their harvesting. In contrast, if they think that there are natural and

[377] There is no perfectly safe vaccine (Chen 1999: 41). However, the number of adverse events in a vaccinated society is vanishing compared to morbidity caused by vaccine preventable diseases in a society without vaccination (Chen 1999: 42). The example is therefore certainly flawed.

[378] Several experiments on decision making with a social context similar to the islanders experiment reveal an interesting detail (Wang 1996). Wang (1996: 46) observed that the most common reason given in post-experiment interviews for choosing the probabilistic outcome is the *norm of fairness*. It was argued that everybody should have an equal chance to survive.

non-man made causes for the shortage they decrease their harvesting. The causes of a certain outcome therefore influence behavior of human beings (Weber, Kopelman, & Messick 2004: 299).

Frames

In decision making experiments providing scenarios that are relevant and socially meaningful reduces reasoning error and decision biases (see also context, Cosmides 1989; Ortmann & Gigerenzer 1997). In rational decision making there is an optimal investment in fact-finding and analysis that leads to rational ignorance (Buchanan 1967: 8).[379] However, the magnitude of such rational ignorance rests on the set of institutions an individual faces if he makes a certain decision. In consequence, "it will be necessary … to examine institutions, not only in terms of the degree of information presented to the ultimate individual participant in collective choice, but also in terms of their predicted ability to foster illusion or false beliefs" (Buchanan 1967: 8).

Decision frames may produce shifts in preferences (Tversky & Kahneman 1981: 453). Prospect theory suggest that people are loss averse (Kahneman & Tversky 1979).[380] People evaluate the *loss* of a given amount of money as more important than the *gain* of the same amount. As discussed in chapter 5.3 social dilemmas appear in two forms: public good problems in which individuals must decide *to contribute* to a resource and common good problems in which individuals must decide *to take* from a resource (Brewer & Kramer 1986: 543). In terms of outcome they are logically equivalent. In both situations cooperative behavior involves the loss of an immediate benefit. However, the two problems refer to different decision frames and are therefore psychologically different. For the public good problem cooperative behavior is connected to a loss while for the common good situation cooperative behavior is connected to a missed gain.[381] In an experiment of Brewer and Kramer (1986: 544) the authors suggest that in a social dilemma framed as a common good dilemma there will be more cooperative behavior than in a public good dilemma. In fact, this turned out to be true (Brewer & Kramer 1986: 548).

In contrast to the reasoning presented by Brewer and Kramer Andreoni (1995b) argues that acting in public good games generates positive externalities and is therefore the positive frame condition. In the common good game cooperative behavior is associated with not producing negative externalities and represents the negative frame condition. The author matches these arguments with real world observation. Most fund raising for charity succeeds in contrast to preserving the commons. Andreoni (1995b: 4) assumes that there is more cooperation in public than in common good situations. Andreoni (1995b: 8) corroborates this observation with experiments. In the public good situation the cooperation rate of the subjects is on the average over ten rounds 34%. In the common good situation the average cooperation rate over ten rounds is only 16%. Although incentives in both situations are similar there seems to be a substantial and significant framing effect on cooperative behavior. The warm-glow of doing well to others in public good games is in this line of reasoning

[379] In many experiments the payoff structure is known to the participants. This is very unrealistic. Generalization of the results beyond the experiment should be discussed critically. Not only that the payoff is known is unrealistic but also that all social and psychological features of human interaction are abstracted. Although this allows to i) investigate the pure structure of the game and ii) to compute a mathematical solution (Buckley, Burns, & Meeker 1974: 293) the results refer to a situation that hardly exist in the relevant real world.

[380] However, loss aversion is not only observable for human beings but also for tufted capuchin monkeys (Chen, Lakshminarayanan, & Santos 2006).

[381] For a short overview over prospect theory and social dilemmas see Schwartz-Shea and Simmons (1995: 90-93).

stronger than the cold prickle of not doing badly to others in the common good dilemma (Andreoni 1995b: 13). The findings of Andreoni (1995b) are supported by an experiment on social dilemmas conducted by McDaniel and Sistrunk (1991). McDaniel and Sistrunk intended to find out whether participants cooperate more in a "take-some" situation with the problem of overuse or in a "give-some" situation with the problem of underprovision. Overuse is the problem of common goods while underprovision is the problem of public goods. The authors found that participants cooperate more in the "give-some" respectively in the public good situation.[382]

The differences reported by Andreoni (1995b) and McDaniel and Sistrunk (1991) on the one hand and Brewer and Kramer (1986) on the other hand may be attributed to differences in context and precise framing. Whatsoever, from this research no definitive conclusion can be drawn whether public or common goods are amenable to cooperative behavior. However, it seems to be clear that framing has an influence even it is not clear in which direction (McDaniel & Sistrunk 1991).

5.5 Enhancing cooperation: Solving social dilemmas

There are basically two solutions for solving social dilemmas (Kondo 1990: 496; Osterloh 2005: 5):[383][384]

i. *Structural solutions* change the payoff matrix until it is for selfish individuals attractive being cooperative (Messick & Brewer 1983: 29-36). A rational egoist cooperates if net returns for doing so are higher than for non-cooperative behavior. To establish such a situation either the costs for non-cooperation are increased (e.g. by punishment, penalties, or sanctions) or returns for cooperation are increased (e.g. by side-payments, rewards, or by activating the shadow of the future). Altering institutions or adding respectively deleting alternatives an individual can choose from may also increase the attractiveness of cooperative behavior (Messick 2000: 233). A variant of a structural solution is to exclude individuals to use a formerly common good if they do not pay a fee. Water consumption (see also chapter 5.3) can be charged by the used cubic meters instead of providing it for free or flat rate billing. By doing so the water consumption is reported to decrease about one-third (Messick & Brewer 1983: 29). In consequence, the problem of overusing ground water may be solved.

ii. *Motivational solutions* focus on the assumption that the maximization of individual monetary benefits is not the only driver of behavior. If individuals think that it is their duty to contribute a fair share the problem of collective action is solved. Sen (1977) describes people who are always selfish as rational fools because this leads to suboptimal outcomes for all involved individuals. For Parsons (1968: 92) normative elements are needed to stabilize social order. Similarly Bierstedt (Bierstedt 1963: 221-222)

[382] It also may be that the observed differences are due to the different context of the two dilemmas. In the "take-some" dilemma participants had to decide on decreasing or increasing milk production which had an effect on milk price. Contribution versus no-contribution to agricultural cooperation was the context of the "give-some" dilemma (McDaniel & Sistrunk 1991).

[383] For Taylor (1987: 21-22) there are also two basic sorts of solution for social dilemmas. Internal solutions do not involve changes in the structure, the game, the possibilities open to the individuals, or the individual preferences. In external solutions the structure, the game, the individuals' possibilities, or the individuals' preferences are changed. In this sense, Taylor's categorization is similar to the one adopted here. For Kollock (1998: 192-) there are motivational, strategic, and structural solutions for solving social dilemmas. Strategic and structural solutions differ from each other insofar as in the former case the fundamental structure cannot be changed while this is possible for structural solutions.

[384] See also Kerr (1995: 34-36) as well as Messick and Brewer (1983: 29-36).

> argues that “whatever order and regularity they {situations in society} exhibit is attributable to the presence of norms to which the participants, in various degrees, conform”.

Structural solutions change the rules of the game to make cooperation more attractive. For example institutions may make an undesirable practice more costly for those who might be tempted to engage in it (Elster 1989b: 149). Related approaches are preferred by economists and human resources managers who believe that compensation policy is the most important part of their job. Motivational solutions focus on activating beneficial preferences of individuals or employees. It is the preferred solution of psychologists.

5.5.1 Changing payoffs: The structural solution

5.5.1.1 Leviathan

As a solution for social dilemmas Hobbes (1998: 114) favors a Leviathan which refers to the gigantic biblical sea monster that cannot be beaten by a *human being*. The Leviathan, a synonym for dictatorship or an all-powerful government, executes his power in the best advantage and in the interest of everybody to overcome the state of nature (Hobbes 1998: 83):

> *Again, men have no pleasure (but on the contrary a great deal of grief) in keeping company where there is no power able to overawe them all.*

Men are led by rational calculation to institute a government over themselves that will keep all of them under control (Orbell & Rutherfo 1973: 384).[385] With a social contract, each man gives up his personal power to one man. Such a structural solution eliminates free choice and therefore the option of acting non-cooperatively (Budescu, Au, & Chen 1997; Komorita & Parks 1995: 196). To secure cooperation there is a deliberate departure from individual freedom (Marshall 1910: 10). With a Leviathan solution it is the sword that secures the covenant and not the word (Hobbes 1998: 111).

The Leviathan has then the capacity to convert a social dilemma into a nondilemma in which cooperation is the dominant choice (Orbell & Dawes 1991: 516). As Ostrom (2000b: 5) argues positive and negative sanction respectively externally implemented incentives are central to public policy textbooks. As she writes the “Leviathan is alive”. Indeed, the argument that cooperation would not be sustained respectively collective goods would not be produced without a Leviathan has always been an important rationale for the existence of governments (Taylor 1987: 1-2).

5.5.1.2 Privileged group

In a privileged group at least one group member has an incentive to provide the collective good even if he has to bear the full burden of providing the good alone (Olson 1995: 49-50). A shipowner may pay for the construction of a lighthouse himself because doing so is less costly than loosing his own ships on the craggy shore. In such a privileged group it is likely that the collective good is produced.

However, there may be a problem if it is known that there are several actors in a group for whom it is rational to produce the collective good even if nobody else contributes. In this situation actors may bargain that

---

[385] This is symbolized in the famous frontispiece of Hobbes’ (1998) Leviathan. It shows a giant king with a sword and a sceptre who is totally composed of many individuals.

somebody else will shoulder the costs of the production (Olson 1995: 50). Even if it is beneficial to produce a collective good alone it is even more beneficial to let others produce the good. One may gamble and assume that the neighbor produces the collective good. If every actor gambles the collective good is not produced. Such bargaining behavior is problematic in one-shoot dilemmas as one cannot revert a non-cooperative choice.

5.5.1.3 Strategic interaction

Cooperation is possible if it serves in the end the strategic goal of selfish egoists to maximize payoff. If the opponents were able to punish defective behavior in subsequent concurrences defection would maximize the immediate payoff but minimize the future payoff. Strategic interaction is based on the threat of retaliation respectively the prospect of future gains.

Tightly related to strategic interaction is the tit for tat strategy in the prisoner's dilemma. Playing tit for tat an actor always cooperates in the first move and the does whatever the other actor did in the previous move (Axelrod 1984: 13). Thereby tit for tat has two characteristics: *Firstly*, defection is immediately punished by defection in the next move and *secondly* defection is quickly forgiven if the opponent cooperates again. A selfish actor who knows that the opponent plays strictly a tit for tat strategy will cooperate in an iterated prisoner's dilemma to avoid retaliation. Tit for tat respectively direct reciprocity can be explained within economic theory and by reference to selfish actors. A reputation for being cooperative may enhance total payoff in the long run.[386] Reciprocity is then not an individual preference and there is no psychological payoff for reciprocity (Sethi & Somanathan 2003: 17). The importance of tit for tat was acknowledged after Robert Axelrod (1984) organized a computer tournament (120'000 moves) of an iterated prisoner's dilemma. The strategy tit for tat handed in by psychologist Anatol Rapoport won the tournament.

An example of strategic tit for tat interaction is the famous Christmas truce on Christmas Eve 1914 during World War I. German and British soldiers visited each other in no man's land, drunk beer, attended church services, exchanged gifts, buried dead persons, and played games of soccer (Weintraub 2001). The German invitation for a truce, namely the proposition "Englishmen, Englishmen. Don't shoot. You don't shoot, we don't shoot", was first not trusted because of heard German treachery (Weintraub 2001: 46). However, communication, particularly singing Christmas carols together and each others hymn, seem to have built up trust (Weintraub 2001: 46-47).[387] The High Commands on both sides demanded firmly the cessation of the truce (Weintraub 2001: 122). In the following years the High Commands commanded artillery bombardments on Christmas Eve to prevent a new truce (Weintraub 2001: 170). The artillery was much less vulnerable to enemy retaliation than the infantry (Axelrod 1984: 81). That is why they were not interested in maintaining the tit for tat cooperation. Troops were also rotated to prevent soldiers from becoming familiar with the enemy (Axelrod 1984: 80).[388/389]

---

[386] However, there is even cooperative behavior in anonymous one-shot experiments (Dawes & Thaler 1988: 189). This may not be explained by tit for tat reciprocity.

[387] Interestingly, there is a related German proverb: "Wo man singt, da lasse dich ruhig nieder, denn böse Menschen kennen keine Lieder" (Own translation: Stay where chants are sung because evil people do not know any songs).

[388] Orders on pain of penalties could only be enforced if behaviour could be directly monitored. This was not easy because "soldiers became expert at defeating the monitoring system" (Axelrod 1984: 82).

[389] After this Christmas truce other small truces were arranged (Axelrod 1984: 78). At one spot "the hour of 8 to 9 a.m. was regarded as consecrated private business and certain places indicated by a flag were regarded as out of bounds by the snipers of both sides" (Morgan 1916: 270-271).

There are some reservations concerning tit for tat respectively strategic interaction as a mean for enhancing cooperation:

- ***Perfect recall:*** Fudenberg and Tirole (1991: 81) conclude that “almost all games in the economic literature are games of perfect recall: No player ever forgets any information he once knew, and all players know the actions they have chosen previously”. The concept of perfect recall in extensive games was introduced by Kuhn (1953: 193) who interpreted it as "equivalent to the assertion that each player is allowed by the rules of the game to remember everything he knew at previous moves and all of his choices at those moves". Perfect recall is based on two independent aspects of memory: i) memory of past knowledge and ii) memory of past actions (Bonanno 2004). The assumption of perfect recall is sound in many cases. There may be situations in which it is not difficult to remember since the amount of information is low. It is mostly possible to remember whether the opponent cooperated in the last round or not. It also may be that the past is memorized collectively (e.g. companies in Axelrod’s Christmas truce example) which makes it more likely that everything is remembered. There are also tools that support remembering such as databases, Google search, newspaper achieves, history books, and diaries. Additionally, there are also individuals with an eidetic memory (see Treffert 1989). However, there are certainly situations in which the assumption of perfect recall is not realistic even only the last move must be remembered.[390]
- ***Detecting defection:*** In Axelrod’s computer tournament defection could be clearly distinguished from cooperation. However, in real life such a distinction is not easy and defection is not always detected. Either actors may camouflage their defection or it is not trivial to detect whether the consequences of a move are influenced by chance, third parties, special circumstances, or deliberate actions. If a team member does not contribute anything valuable to a brainstorming it may be that i) he did not even tried to find an idea or that he ii) did so but unfortunately nothing came to his mind. If a cooperative move is wrongly assumed to be a defective move a long-running cycle of retaliation may start if both actors play a tit for tat strategy. Furthermore, it is reasonable to assume that there is a threshold for the detection of non-cooperation (Schuessler 1990, see also chapter 6.1.3). If continuous instead of discrete contributions are possible an individual may be just a little bit uncooperative which in turn is difficult to detect. It is reasonable to assume that in many real world situations decision making is not binary but continuous. In a simulation of prisoner's dilemmas such detection thresholds had negative implications on the stability of cooperation and led almost instantaneously to the collapse of cooperation (Schuessler 1990: 492). During the cold war it was thought that the concept of MAD (mutual assured destruction) ensures peace. However, important problems were false positives. Early warning systems were and are not perfect in ruling out false reports of hostile attacks.[391] For example in November 9, 1979 training tapes that simulated a massive Soviet strike were accidently inserted into the computers that processed data from United States early-warning radars (Forden, Podvig, & Postol 2000). In consequence, it was falsely assumed that the United States were under attack. If

[390] In 1990 the German Democratic Republic ceased to exist. A study about adolescents reveals that knowledge about the German Democratic Republic vanishes (Flohr 2008). About 40% of the adolescents from East Germany state that there was less environmental pollution in the German Democratic Republic (GDR) than in the Federal Republic of Germany. Only 38% of these adolescents know that the chief executive body of the GDR was not freely elected.

[391] Faulty radar signals, computer malfunction, or garbled electronic communication could cause such a false alarm (Carnesale & Glaser 1982: 72).

actors play in such a situation a tit for tat strategy a falsely interpreted move starts an infinite spiral of retaliation.

The difficulty to detect defection is especially high if tacit knowledge is involved. In the resource based view (Barney 1991; Prahalad & Hamel 1990) resources are especially important if they are difficult to imitate or substitute. Such resources are mostly based on tacit knowledge. It can therefore be concluded that strategic interaction as a solution is not very powerful in the cases in which valuable resources as defined in the resource based view are involved.

- ***Meeting again:*** A prerequisite that strategic interaction leads to cooperation is that actors must have a reasonable chance of meeting each other again. Additionally, the future must be sufficiently important relative to the present (Axelrod 1984: 126-132). If the shadow of the future is not large enough actors will not be deterred by retaliation. Rotating troops in World War I prevented them to fraternize. This shortened the time in which strategic interaction could be built up as well as the time for reciprocation.

  A further problem is that the actors have to identify each other (see also perfect recall). In some instances such as in small villages this is not a problem while in global cities this is more troublesome. It is also possible to camouflage an identity; a problem virulent in online interaction. Producing an email at Yahoo with a non-existing name is no problem.

- ***Dependence on opponents' strategy:*** A seldom-recognized fact is that tit for tat had only the best *average score over the entire tournament*. Some strategies such as "defect always" performed better in the individual encounter with tit for tat.[392] In a tournament with persistent defectors tit for tat looses. It is therefore clear that the advantage of playing tit for tat depends on the strategies of the others players.

- ***Backward induction:*** A central concept in game theory is subgame perfection (Camerer 1997: 180-181). A pattern of behavior is subgame perfect if actors i) consider in their decision every possible subgame that could be reached later in the game tree, ii) guess what the actors do in these subgames, and iii) use these guesses in deciding what to do. This requires actors to induct future events backward to the present (Luce & Raiffa 1957: 97-102). If the number of rounds played is known the players know that a rational egoist will defect in the last round (Pettit & Sugden 1989: 169-170). There will be no future possibility for retaliation. Because of this it does not make sense to cooperate in the second last round because the opponent will defect in the last round anyway. In consequence, it is not rational to cooperate in the third last round and so forth. In Axelrod's second computer tournament the length of the games was determined probabilistically. There was a 0.00346 percent chance that the game was to be terminated to rule out such end-game effects (Axelrod 1984: 41-42). That there is no known end seems to be realistic in most real life cases but certainly not in all.

Strategic interaction is not only based on direct reciprocity (A helps B because B has helped A before) but also on indirect reciprocity (A helps B because B has helped C before, Rutte & Taborsky 2007: 1421). Indirect reciprocity involves reputation mechanisms. This obsoletes the necessity of meeting again discussed above. However, actors do still need very good cognitive abilities for making strategic interaction work properly.

[392] An actor playing tit for tat starts cooperatively and then always retaliates. An actor always defecting does so as well in the first round. Because of the very first round the defector performs better than an actor playing tit for tat.

5.5.1.4 Privatization

In an early article Gordon (1954) discusses the problem of fishery commons. He writes (Gordon 1954: 135):

> *There appears, then, to be some truth in the conservative dictum that everybody's property is nobody's property. Wealth that is free for all is valued by none because he who is foolhardy enough to wait for its proper time of use will only find that it has been taken by another. The blade of grass that the manorial cowherd leaves behind is valueless to him, for tomorrow it may be eaten by another's animal … the fish in the sea are valueless to the fisherman, because there is no assurance that they will be there for him tomorrow if they are left behind today.*

Gordon (1954: 130-131) therefore concludes that because "in the sea fisheries the natural resource is not private property" the fisher which is a rational egoist will catch as many fish as possible. However, if common goods are privatized the corresponding social dilemma disappears.

Ellickson (1993: 1335-1344) reports the history of three pioneer settlements in the United States. All settlements started with common property but finally privatized it. Ellickson argues that common property had the main advantage of risk spreading. The pooling of risks was necessary because hazards such as starvation, tribal raids, and diseases were prevalent. With the time going on these risks scaled down. Consequently, the common property was privatized because thereby productivity could be increased.

However, privatization is sometimes not possible or not wise. Privatize the ocean is problematic because fish will not acknowledge borders. Lobsters for example are found in different places over the annual cycle and from year to year (Acheson 1987: 61). In warm months they are found in shallow water while in cold months lobsters are found in deep water. A rational egoist does then not limit his harvesting because the spared fish are of no use in the future. Creating profit centers may hinder the transfer of tacit knowledge (Osterloh & Frey 2000). There is a priori no incentive to share knowledge between profit centers because they want to maximize their own and not other profits. Furthermore, other profit centers may turn into competitors. At the same time tacit knowledge can hardly be monitored and it is therefore difficult to regulate such knowledge transfer. As the resource based view argues such tacit knowledge may be the source for sustained competitive advantage (Barney 1991; Prahalad & Hamel 1990). Not sharing knowledge respectively installing profit centers may therefore have adverse affects for commercial firms.

Generally, creating synergies is what makes it beneficial to organize individuals in firms and not depend on market transactions. To create synergies seems to get more important with technological advancement because with the specialization of tasks the interdependency between these tasks increases as well (Simon 1991: 42). Nevertheless, building up synergies is accompanied by teamwork and social dilemmas. Indeed, the need for collective action is what all organizations have in common (Arrow 1974: 26). Privatization of collective goods may therefore have adverse effects.

5.5.1.5 Institutions

North (1997: 3-4) argues that in a society formal or informal institutions are the rules of the game that constrain human interaction (see also chapter 5.5.1.1). They define and limit the set of individual choice. According to Buchanan (1967: 5) "common sense indicates that the institutions through which costs and benefits are presented to the private citizen may influence his decision." Not only individuals but also organizations take advantage of the opportunities created by institutions (North 1997: 7).

Institutions are necessary to mobilize collective action because no "decentralized pricing system can serve to determine optimally these {optimal} levels of collective consumption" (Samuelson 1954: 388).[393] By providing a structure to everyday life institutions also reduce uncertainty.

According to Elster (1989b: 147) institutions keep society from falling apart, provided that there is something to keep institutions form falling apart themselves. This note points to one of the big problems of institutions. It may be that institutions have developed by accident, by survival in a social evolutionary process, or by technological necessity (Buchanan 1978: 364). However, some institutions need deliberate collective action for maintenance and generation.

5.5.1.6 Side-payments or positive selective incentives

Olson (1995: 51) argues that in latent groups[394] only selective incentives will stimulate a rational individual to act in a group oriented way. Labor union for example may coerce members to join their organization (Olson 1995: 68) or provide insurance services (Olson 1995: 72). While the former practice is a negative selective incentive the later is a positive selective incentive.

According to Olson (1995: 60) economic and monetary incentives are not the only positive incentives. Human beings are also motivated by the desire to win prestige, respect, friendship, and other psychological and social objectives.[395] Additionally, there are erotic and moral incentives (Olson 1995: 61).

Positive selective incentives are private goods only accessible for individuals who cooperate or act in the interest of the group (Olson 1995: 51). Olson's (1995: 72) classical example is the aforementioned exclusive insurance for union members.

5.5.1.7 Sanctions or negative selective incentives

For John Stuart Mill (1879 chapter 3) the motivation to follow moral principles are *internal* or *external* sanctions. Sanctions and punishment respectively negative selective incentives discussed here are clearly *external* sanctions.[396] By the punishment of defectors the costs for defection rises that makes it more likely that the net payoff for cooperation gets positive. Negative selective incentives are the counterpart of positive selective incentives (Olson 1995: 51). They make cooperative behavior in contrast to defection more profitable. An example for negative selective incentives is violently deterring scabs from crossing the picket

---

[393] Mechanisms to determine optimal levels of collective consumption such as voting or some sort of signalling of preferences have the problem that "it is in the self interest of each person to give false signals, to pretend to have less interest in a given collective consumption activity than he really has" (Samuelson 1954: 389). It is a dominant strategy to hide the magnitude of interest for a public good because an individual cannot be excluded from its benefits independent on the signalled preferences. "Utopian voting and signalling schemes", as Samuelson (1954: 389) calls it, are therefore related to motivational solutions (see chapter 5.5.2).

[394] A latent group is a large group in which nobody is significantly affected by a third person contribution to a public good (Olson 1995: 50).

[395] In Olson's (1995) main work such non-monetary incentives seem to be of no special importance. Interestingly, in a less known contribution of Olson (U.S. Department of Health Education and Welfare 1969) he puts more weight on non-monetary factors (Pluck 1969; Taeuber 1970).

[396] Mill (1879, chapter 3) defines internal sanction as "whatever our standard of duty may be" and as "a feeling in our own mind; a pain, more or less intense, attendant on violation of duty, which in properly cultivated moral natures rises".

line (Olson 1995: 71-72). In the famous miners' strike between 1984 and 1985 in the United Kingdom scabs were physically threatened (Greatrex 2004).[397]

Mutual coercion is also the solution suggested by Hardin (1968: 1247) to avoid the tragedy of the commons. Hobbes solution to the problem of social dilemma is that people hire an agent, specifically a monarch, whose duty is to punish anyone breaking rules. People do not like being supervised by a tyrannical Leviathan. Everybody prefers that he is not under the rule of this Leviathan but all others. However, because there is only the choice between everybody or nobody under the rule of the Leviathan people choose tyranny over anarchy as the lesser of the two evils.

Yamagishi (e.g. 1986; 1988c) studied the provision and use of punishment in several experiments. Indeed, if defection is punished there is more cooperation (Yamagishi 1988c: 38).

5.5.2 From payoffs to utilities: The motivational solution

Many people would not to burgle a house even if they knew that it is very likely to get away with it, that punishment was in either case minuscule, and that no friend will ever found it out. On the average 80% of the citizens in western democracies vote in national elections (Freitag 1996) although the probability that it will have any influence on the result is tiny.[398] In both examples people reject a larger payoff for one that is smaller. The burglars could get rich and voters could save time. Apparently, these individuals have utilities that are not related to external payoffs (Dawes 1980: 175).[399] In such a case social dilemmas are transferred to an assurance game (Ostrom 2000b: 8).

5.5.2.1 Altruism and pro-social behavior

Although many social values have been identified competition (maximization of payoff difference), cooperation (maximization of joint payoff), and individualism (maximization of own payoff) have received the greatest support in research on social values (Komorita & Parks 1995: 197-198). Altruism as the maximization of other's payoff has received less attention.[400/401] The term altruism as an opposite of egoism was coined by Comte (2000) although its basic has been long discussed before under other terms such as benevolence, charity, compassion, and friendship (Batson & Shaw 1991: 108).

Sometimes altruism and cooperation have been merged to pro-social behavior. In other cases altruism and pro-social behavior have either been used interchangeable (see the different book sections in Wispé 1978) or with subtle differences (see for example Batson & Shaw 1991: 109). Here it is assumed that pro-social behavior includes altruistic as well as cooperative motivation.

---

[397] In fact, people were killed such as taxi driver David Wilkie who drove scabs to work (BBC News 2006a).

[398] That holds even true in very close elections as for example in the United States presidential election of 2004. The electoral votes of the swing state Ohio finally decided which candidate won presidency. In this state the difference between the two opponents was 118’601 votes (Federal Election Commission 2005: 6).

[399] The argument that all behaviour is selfish because it reflects utilities seems to be trivially true as long as i) one equates utility with pleasure and happiness or prevention of pain and unhappiness and as long as ii) one beliefs that “nature has placed mankind under the governance of two sovereign masters, pain and pleasure” (Bentham 1789 chapter 1). It is sometimes discussed whether altruism is based on genetic evolution (Margolis 1984: 26-35; Wilson 1978; Wilson 2005) and not on deliberate decisions.

[400] For a discussion of the definition of altruism see Batson and Shaw (1991).

[401] For a critical view on altruistic behavior see Maner et al. (2002).

Individuals engage in pro-social behavior that it is costly to themselves but benefit others. This is confirmed by experiments and field studies in which a certain fraction of people engages in such behavior (Benabou & Tirole 2005: 1; Charness & Rabin 2002: 817). In anonymous dictator games proposers donate on average 20% to 30% of the sum to responders (Camerer 1997: 169). Strategic interaction cannot be the cause for this result because the responder cannot reject the offer and therefore retaliate. Motivation that takes the outcome of others into account heavily influences collective action. The drift of individuals to maximize pecuniary rewards may be transformed into cooperative moves (Fehr & Schmidt 1999; Rabin 1993). If pecuniary motivation as well as social motivation is taken into account a prisoner's dilemma can be transformed into a coordination game (von Hippel & von Krogh 2003: 217).

5.5.2.2 Social norms

López-Pérez (2006: 23) argues that social norms are an important factor to explain observed cooperation in game theory.[402] A social norm is a behavioral rule that is accepted by different members of a group (Thibaut & Kelley 1959: 147). It is a societal expectation that governs behavior in social situations (Bierstedt 1963: 222). It is thereby not enforced by legal sanctions (Posner 1997: 365). A *social* norm differs from a *personal* norm insofar as it is administered by other's than the individual which is subject of the norm (Kerr 1995: 33) respectively if the norm is at least partly sustained by the approval or disapproval of other people (Elster 1989b: 113).[403/404] Social norms are not oriented towards an outcome but towards behavior.[405] A social norm urges to do X but not Y. At the most general level norms are valuable because they reduce the need for the permanent exercise of direct personal influence. The community as a whole looks for conformity. This may reduce costs, uncertainties, resistances, and conflicts. Social norms also contribute to group solidarity and group cohesiveness (Thibaut & Kelley 1959: 139).

Group members follow a norm mainly because they consider it themselves as right, true, or valid.[406] They feel some obligation to adhere to the rules. However, nonadherence to norms and rules result also in the exercise of influence to restore conformity. Therefore, it is difficult to distinguish between behavior based on social norms and behavior that help others but is basically "motivated by the expectation that doing so will bring social reward" (Blau 1964: 17). Insofar social norms are connected to negative or positive selective incentives. At the same time, conformity to social norms cannot exclusively be explained by sanctions. Sanctioning people who do not conform to a norm is a norm itself. Individuals not willing to punish others may be punished themselves and so forth (see also chapter 5.6 on second order social dilemmas). The punishment of the defectors of a social norm is therefore based on a higher order social norm. This leads to an infinite regress.

[402] For an overview on how norms affect behavior in and of corporations see Moch and Seashore (1981).

[403] Social norms may be internalized and therefore get personal norms (Kerr 1995: 33). Internalized norms are followed even when violation will not be detected and in consequence sanctioned (Elster 1989b: 119).

[404] For a discussion of the different meanings of the term "social" in ordinary English and American English see Weitman (1978: 229)

[405] Rational choice is instrumental because it is guided by the outcome of behavior (Elster 1989b: 22). Behavior is then valued and chosen not for itself but as an efficient mean to a further end.

[406] In a typology of Krebs (1997: 359) individuals adhere to norms if i) it is costless (in contrast to a violation of norms), ii) it is immediately rewarding because it permits coordination (such as walking right in a crowded walkway), iii) it is rewarded, or if it is iv) desirable per se. In a different typology Posner (1997: 365-367) argue that there are four reasons to obey norms: i) doing so is constitutive for advantageous transactions (to enjoy playing chess one has to play by the rules), ii) there are emotional benefits, iii) they are enforced by expressions of disapproval or ridicule, and iv) the norms are internalized and obeyed out of a sense of guilt or shame.

Therefore, there must be something what Aristotle would have called an unmoved mover, a prime mover, or a first mover (see e.g. Aristoteles 1994: 318; for a commentary see Ross 1958: 372-382; for a discussion see Wolfson 1958).[407] Consequently, at least some norms must be adhered to for motives other than the fear of being sanctioned.[408] Additionally, most sanctions related to norms violation only work if individuals feel at least some minimal amount of shame. Otherwise, disapproval glances will not work in an anonymous big city. It can be concluded that behavior based on social norms and sanctions are theoretically not distinguished neatly but that there is nevertheless a fuzzy difference.

Norms can be described by different approaches. Thibaut (see also Margolis 1990: 821; 1959: 126) states that norms can be discussed according to their *social processes and mechanisms* (e.g. pressure, obligations), according to their *effects* (e.g. behavioral uniformity, shared frames of reference), or according to their *functions* (e.g. provision of support for opinions, facilitation of group achievement).

Because norms are learned in social milieus they vary substantially across cultures, across individuals within a culture, within individuals across different situations, and across time within any situation (Ostrom 1998: 9). Yet there are for Kerr (1995: 36-42) three universal norms that may regulate behavior in social dilemmas:[409]

- ***Reciprocity norms***: Reciprocity norms prescribe that for a received benefit an equivalent benefit should be returned. Following such a norm Alice cooperates if Bob did so as well. Ostrom (1998: 10-15) emphasizes the special importance of reciprocity in social dilemmas.[410] According to Fehr and Fischbacher (2004: 186) human cooperation is partly based on social norms of reciprocity respectively conditional cooperation. Human beings may have a preference for reciprocity (Sethi & Somanathan 2003: 17-23). However, according to such a norm reciprocating another's cooperative behavior is only required if the intention was benevolent. Cooperative behavior based on stupidity, confusion, or a deceptive strategy needs not to be reciprocated.
- ***Commitment norms*** (see also Biel 2000: 34): Commitment norms prescribe that if an individual makes a promise his or her action should be to some extent consistent with his or her words. Communication fosters cooperation (see chapter 5.4.2.1.2). An explanation for this observation is that the possibility of communication allows making a commitment that in turn has to be hold.
- ***Equity norms:*** Equity norms apply to the allocation of resources among group members (see chapter 5.4.2.1.2). Payoffs should be distributed in proportion to contributions, inputs, or costs. In consequence, an individual who stops contributing to a public good should stop being a user of the public good if he follows equity norms.

[407] While rationale choice theory can hardly escape vicious circles (Harsanyi 1977: 113) or infinite regress (Schelling 1980b: 87) theories that rest on social norms suffer from dogmatically installed principles (respectively *unmoved movers*). One may therefore criticize this argument by noticing that an infinite regress is substituted by installing dogmatically a basic principle respectively *an unmoved mover* (Albert 1991: 15). This critic is certainly sound. It seems to be questionable whether it is possible at all to escape what is called the Munchhausen Trilemma (Albert 1985: 18). The trilemma forces to choose between i) an infinite regress, ii) a logical circle, or iii) a dogmatically installed foundation respectively *an unmoved mover*.

[408] There are probably some cases in which social norms are the outcome of an anonymous process of social selection during which communities with useful norms eliminate other communities that do not have these social norms (Corneo & Jeanne 1997: 334). In other cases, however, deliberate investments in social norms have to be made.

[409] For a description of different types of norms see Bierstedt (1963: 224-226).

[410] However, van der Heijden et al. (1998: 31) found for social dilemmas no systematic relation between the gift received and the gift returned. They conclude that there were very few signs for reciprocity.

Taylor (1982; 1987) investigates the conditions under which social order can be maintained without a Leviathan. He concludes that communities are very important to sustain order (Taylor 1982: 2). The proper functioning of communities is dependent on shared beliefs and norms, direct and complex relationships between community members, and reciprocity (Taylor 1982: 26-30).[411] Norms may therefore not only lead to cooperation directly but also indirectly because it fosters community coherence.

However, it must be clearly stated that norms are not per se beneficial for society or the individuals itself. The norms of the Camorra or the Cosa Nostra may be highly efficient in respect to these organization's goals but destructive for society. There may also be too many norms or too complex norms so that proper behavior gets inefficient. The behavior of an individual may also be totally guided by norms without responding to the outside world or what is demanded by the unique situation (Thibaut & Kelley 1959: 140). In such cases norms may get pathologic. A corresponding example may be cults that dominate individual behavior with the help of norms to the disadvantage of the individuals.

5.5.2.3 Is all human behavior egoistic?

Advocates of universal egoism argue that all human behavior is in its end egoistic because it gives at least a good feeling and prevents a bad conscience (Batson & Shaw 1991: 107; see Dawes & Thaler 1988: 192).[412/413] In fact, the concept of utility refers to the amount of welfare an agent derives from an object or an event (Ross 2006). Welfare refers to a normative index of relative well being justified by reference to a background framework. The welfare of individuals refers to their own implicit or explicit judgment. Utility is therefore a measure of subjective psychological fulfillment. In this sense all human behavior is egoistic.

For Bateson and Shaw (1991: 109) behavior is altruistic if the relieve of other's suffering is the ultimate goal and if self-benefits are unintended consequences. In egoistic behavior the relieve of other's suffering is an instrumental goal and self-benefits is the ultimate goal.

The discussion whether altruism or the like and egoistic behavior is in its end rooted in the same concept has a long philosophical tradition (White 1995: 258). In philosophy there are mainly two interpretations on this issue: a *Kantian* and a *Hegelian* interpretation.[414] The *Kantian* view is dualistic as it regards ones own good and the good of other persons as two different types of considerations (White 1995: 259). These considerations are independent from each other and one end cannot be derived from the other. Further, the two types of considerations might conflict with each other because they sometimes recommend different courses of action. In that case, a person has to follow one or the other course of action. In the *Hegelian* view this dualism is denied (White 1995: 260). It is argued that there is no conflict between the good related to one's own person and the good related to others. Happiness is therefore *formally* egoistic because it represents one's ultimate aim as a state of oneself (White 1995: 262). This does not imply that happiness is *substantively* egoistic because

[411] Generalized reciprocity is vague, diffuse, and not conditional upon reciprocation (Taylor 1982: 29-30). In balanced reciprocity there is a direct and conditional exchange of goods. Reciprocation must be made then simultaneously or shortly after receipt. Negative reciprocity is the attempt to get something for nothing or in other words theft, haggling, or gambling.

[412] For a discussion about the topic see also Rocha and Ghoshal (2006).

[413] Sen (1977: 326) distinguishes between sympathy and commitment. In the former case the concern for others directly affects one's own welfare. If it makes you sick that other people suffer it is a case of sympathy. If something does not make you feel worse but you think that it is wrong and you start reacting it is a case of commitment.

[414] These competing views do not originate from Kant or Hegel but are associated with them (White 1995 259).

one's good can include ethical or altruistic components. Following this view John Stuart Mill (1879 chapter 4) elaborates on virtue, money, fame, and power:

> *In being desired for its own sake it is, however, desired as part of happiness. … The desire of it is not a different thing from the desire of happiness, any more than the love of music, or the desire of health. They are included in happiness. They are some of the elements of which the desire of happiness is made up. Happiness is not an abstract idea, but a concrete whole; and these are some of its parts.*

The question whether all behavior can be discussed within the utility framework or not is left to others to answer. However, there is in either case some competition between different means of action. Buying a cup of tea depletes the resources necessary to buy a cup of coffee. In the case of social dilemmas the appropriate course of action is either self-interested or related to social interests (Baden 1998: 51). Therefore, it seems appropriate to treat pro-self and pro-social behavior as something distinct.

5.6 Structural or motivational solutions?

In the following, the problems of structural as well as motivational solutions are discussed. In short, it must be acknowledged that both types have its pitfalls. It will be argued that solving the problem of collective action will in most cases include a multiple-step process consisting of structural as well as motivational solutions.

5.6.1 Structural solutions are not without pitfalls

According to Olson (1995: 51) in latent groups[415] only selective incentives will stimulate a rational individual to act in a group oriented way. Labor unions for example may coerce workers to apply for membership (Olson 1995: 68) or provide insurance services (Olson 1995: 72). However, either framed as negative or positive incentives it is not clear why and by whom the incentives are paid or exerted. Let us assume that a few labor union leaders benefit from the existence of such an organization because it pays their salaries. For these leaders it is then worthwhile to organize the proper functioning of the labor union. These leaders may either force workers to join the union or provide positive selective incentives to do so. Nevertheless, if the labor union leaders may force the workers to join the organization against their will why do they bother at all spending money for labor union work? Similarly, if the workers are only interested in getting a selective incentive, e.g. a proper insurance, why should an organization that wants to maximize its profits not only focus on selling these insurances?

Let us assume that a labor union cares about lobbying and spends scare resources for this task. To get members the labor union provides insurance. Let us further assume that commercial firms are as well able to provide insurance for workers. Because the commercial firms do not have to pay for the lobbying the insurance fee will certainly be lower than the member fee of the union (that includes a similar insurance coverage). Workers only interested in the selective incentive would unanimously choose to buy the insurance from the commercial company because it were cheaper. The same reasoning may be applied for negative selective incentives. Paying the protection money to the mafia is ceteris paribus cheaper than paying the union member fee because the later has additionally to pay for the lobbying. The fact that leaders benefit from sustaining an organization is therefore not sufficient to explain why the public good is produced. In short, the

[415] See Footnote 394.

question remains why nobody only produces the selective incentive without paying for the lobbying. One may argue that the selective incentive can only be produced as a by-product of lobbing. Indeed, there are probably very special cases in which this is true. However, for Olson's examples of coercion and insurances and probably for most other examples this is clearly not the case.

That structural solutions are not easily implemented can be recognized by the observation that many "pay for performance" systems do in fact not pay for performance. After a review of different studies Lawler (1971: 158, cited in Baker, Jensen, & Murphy 1988: 595) concludes that

> *their evidence indicates that pay is not very closely related to performance in many organizations that claim to have merit increase salary systems. . . . The studies suggest that many business organizations do not do a very good job of tying pay to performance. This conclusion is rather surprising in light of many companies' very frequent claims that their pay systems are based on merit. It is particularly surprising that pay does not seem to be related to performance at the managerial level.*

In the following problems of structural solutions are discussed (see also Marwell & Oliver 1993: 8):

- ***Second order social dilemmas:*** The installation of structural solutions alters a social dilemma into a second order social dilemma (Biel 2000: 28; Buckley, Burns, & Meeker 1974: 293; Elster 1989a: 41; Heckathorn 1996: 253; Kollock 1998: 206; Oliver 1980; Ostrom 2000b: 3-4; Yamagishi 1986).[416] Somebody has to pay for the monitoring and the investment in selective incentives (see also the example of labor unions above). However, doing so is a social dilemma itself. Individually it is rational to let others to pay for the selective incentives that in consequence may lead to the situation that nobody pays. As Posner (1997: 366) argues norm enforcement is costly and therefore subject to a free rider problem.

  Ostrom (1997: 95) describes how the second order social dilemma can be solved. She argues inter alias that the costs associated with monitoring and sanctioning most be low. In long enduring common pool resources monitoring was often a by-product of using the commons (Ostrom 1997: 96). Buckley (1974: 293) argues that solving higher order social dilemmas depends usually on social solidarity, social relations among actors, cultural understandings, and social norms.

- ***Costs:*** A further problem of structural solutions is the corresponding costs (Hechter 1984).[417] Even if the problem of collective action and the second order social dilemma is somehow solved the benefits of a collective good are diminished by the costs for providing it. Sometimes the costs of rewarding or punishing people exceed the gains a society derives from having everyone to cooperate (Dawes 1980: 175).

---

[416] Interestingly, empirical evidence shows that individuals contributing to a first order social dilemma do not necessarily contribute to a second order social dilemma (reported in Kollock 1998: 206).

[417] An example that extrinsic motivation to ensure behavioral compliance may be costly is the golden handshake Robert Nardelli received for resigning as a chairman and CEO of Home Depot. During his term as a CEO the stock price of Home Depot declined 6% while the one of the archrival Lowe increased by more than 200% (Grow 2007). Nardelli received according to the separation agreement approximately $210 million. In exchange Nardelli has agreed to behave in the interest of the company. This means to not "compete with the Company *for one year*, not to *solicit* employees or customers of the Company for four years and *other restrictive covenants*" (Home Depot 2007, emphasis added). A cheaper solution to the problem of proper behavior after resignation would have been work ethics and interest in the well-being of the company.

Structural solutions depend basically on supervision because one has to decide who has to be punished or rewarded. The problem of supervision is that it may be prohibitively costly to do so (see also Alchian & Demsetz 1972: 780; and Sen 1977: 333-335). To fine a person littering a park may work during daytime but it does not work easily in the night where nobody is around to observe the misdemeanor. The citizens may agree to pay night watch men to prevent littering in the night. However, the negative utility for paying the night watch men respectively increased taxes reduces the total net utility from having a park free of garbage. In consequence, a collective good that is only provided if there are sufficient selective incentives may in some cases be so expensive that having it produced is comparable to a pyrrhic victory. One may find many corresponding real world examples.

- ***Competition and selective incentives:*** A hardly discussed problem is why organizations that provide a collective good by offering selective incentives are not outcompeted by commercial firms that provide the very same selective incentives. These firms could sell their services and products certainly cheaper because they do not have to pay for the collective good. It may be argued that only somebody that produces the collective good may produce the selective incentives as a by-product. This seems to hold true for very special cases but is not universally true. Olson's (1995: 68 and 72) examples of coercion and insurance services do hardly depend on lobbing.

- ***Imperfect rewarding and punishment:*** Structural solutions may produce problems of its own or lack efficiency. To effect cooperative behavior one has to specify clearly and depletive which behavior is rewarded and which behavior is punished. This prerequisites a tremendous knowledge about the problem and the connection between all influence factors. If rules respectively rewarding and punishment are imperfect actors will find a way to evade cooperation in order to receive a higher payoff.[418] Measured performance can be improved either by changing the performance or by changing the accounts of performance (March 1981: 568). If hospitals are rated by measures of quality such as the Standardized Mortality Ratio they may for example artificially bias this benchmark by transferring critically ill patients to other acute care hospitals (Kahn, Kramer, & Bubenfeld 2007). Measuring hospital quality by the time it needs until a patient gets treatment after he crossed the hospital doorstep may be flawed, too. As happened in the United Kingdom a hospital may let patients wait in the ambulance right in front of the hospital until the medical team is ready. Baker, Jensen and Murphy (1988: 597) state:

  > *We believe that careful examination of the criticisms of monetary pay-for-performance systems indicates not that they are ineffective but rather that they are too effective: strong pay-for-performance motivates people to do exactly what they are told to do. Large monetary incentives generate unintended and sometimes counterproductive results because it is difficult to adequately specify exactly what people should do and therefore how their performance should be measured. Moreover, merit-pay systems encourage employees to spend effort lobbying about both the specification and application of the system to measure and evaluate output.*[419]

[418] However, this means at the same time that pay for performance may be a successful way for improving organizational success if the task can easily be monitored. Lazear (2000) found for example a huge increase in productivity for auto glass installation after the Safelite Glass Corporation switched from a hourly wage rate to a piece rate compensation.

[419] In relation to this citation Baker, Jensen and Murphy (1988: 595) report the study of Medoff and Abraham (1980) in which the pay for performance system of two large US manufacturing companies were researched. In both companies about 95% of the employees were classified as *good* or *outstanding* (respectively superior) employees (Baker, Jensen, &

In fact, in accounting there is a divergence between the rule-based US-GAAP and the principle-based IAS accounting standard (Benston, Bromwich, & Wagenhofer 2006; Shortridge & Myring 2004). In the rule-based approach there are specific details to address as many potential contingencies as possible. In contrast the principle-based approach provides a conceptual basis to follow instead of a listing of detailed rules. If accountants are in doubt about rules they are urged to comply with the principles. The huge problem of the rule-based approach is that if the rules are not clear and depletive there is a legal way to offend its intent. In the Enron Corporation failure Arthur Anderson seems to have designed instruments that met the technical requirement of US-GAAP while violating its intent (Benston, Bromwich, & Wagenhofer 2006: 166).[420] In Hamel's (2000) appraisal of Enron's innovation style he concludes that "Enron understands that if you want people to create wealth, you have to share the wealth – no just with a few privileged executives, but with successful revolutionaries at all levels".[421] However, Hamel did not consider that "rewarding entrepreneurs on a scale commensurate with their accomplishments" rest on the assumption that faked and virtual accomplishments can be identified and that the control instances are interested to punish deceit. If everybody in an organization is highly dependent on mutual rewarding such prerequisites may be problematic.

In the subprime mortgage crisis 2007/2008 the UBS lost until March 14, 2008 approximately 62.6 billion dollars in market capitalization (Blundell-Wignall 2008: 6). In a shareholder report the UBS presented the key findings related to the causes of the losses. It was identified that four of the overarching causes were related to compensation and incentives (UBS 2008: 41-42):

- ***Structural incentives to implement carry trades:*** "… employee incentivisation arrangements did not differentiate between return generated by skill in creating additional returns versus returns made from exploiting UBS's comparatively low cost of funding in what were essentially carry trades."
- ***Asymmetric risk / reward compensation:*** "The compensation structure generally made little recognition of risk issues or adjustment for risk"
- ***Insufficient incentives to protect the UBS franchise long-term:*** "Essentially, bonuses were measured against gross revenue after personnel costs, with no formal account taken of the quality or sustainability of those earnings."

These findings may lead to the following conclusion: If even the "world's biggest manager of other people's money" (Economist 2007) does not succeed to manage pay for performance properly it is not a trivial thing to do.

Murphy 1988: 595). The premium for being an outstanding (respectively superior) rather than a good employee is 2.5% in one company and 1.8% in the other company. For a monthly salary of 3000$ the premium for being an outstanding employee is 54$ respectively 75$.

420 However, the principle-based approach is not without problems (Shortridge & Myring 2004: 36). Lacking precise guidelines comparability among different organizations and consistency that are basic to accounting are reduced.

421 One has to read the first edition (Hamel 2000) published before the Enron bankruptcy because in the second edition (Hamel 2002) after the Enron bankruptcy there are only four instances left in which the term Enron is mentioned. In the second edition Enron is almost completely deleted. Hamel (2002: 14) explains the collapse of Enron with a "swashbuckling finance function". After having praised Enron's innovation systems in the first edition that much this austerity may be considered questionable. Additionally, it seems to be a missed chance to discuss pitfalls and disadvantages and not only advantages of innovations systems.

Another informative example how the intent of a regulation is evaded without infringing it technically or legally is fishery management. In fisheries input restrictions and limited entry produced strong incentives to increase the use of unregulated inputs (Townsend 1995: 39). Quotas created the problems of races to exhaust these quotas. Another problem of quotas is that within the same species the value of a fish may heavily depend on size, sex, or, physical condition (Clucas 1997: chapter 4). In anticipation of catching fish of higher value some part of the catch is discarded to the sea to not reach quotas what is called high grading. A similar problem can be observed in multi or mixed species fisheries. A ship may reach the total allowable catch for a particular species while there is still unfulfilled quota of another species. The ship will continue fishing but discard fish for which the quota has been filled. At best these different regulations were not successful in managing fishing.

There are also adverse effects as unintended by-products of rewarding. An example is the evaluation of Members of the Swiss Parliament. Newspapers evaluate yearly how influential these members are. One item of the measurement system is the number of requests to the government (e.g. motion, postulate, or interpellation) they made. It is assumed that the more requests a member made the more influential he is. In consequence, this evaluation seems to have increased the number of requests (National Council Member 2007) in order to get a higher evaluation. Indeed, next to many interesting requests there are also requests with questionable value.[422] In January 2007 exactly 1300 motions, postulates, or interpellations were outstanding (Swiss Parliamentary Service 2007). While in the last session in Flims 2006 258 requests were submitted only 209 were completed.

- ***Multi-tasking:*** Tightly related to imperfect rewarding is the problem of multi-tasking (Holmström & Milgrom 1991). In most cases employees have to accomplish many different tasks. While some of these tasks are easy to monitor others are not. In this situation a rational egoist will optimize his compensation by concentrating his efforts on the tasks that are easy to monitor.

  It could be argued that in many cases the tasks that are the easiest to monitor are the least valuable. An example may be the evaluation of Swiss military captains: A visiting brigadier general may judge very fast whether the company parades neatly or not. However, the proficiency in parading does not tell too much about combat readiness. At the same time an ambitious captain should nevertheless invest a lot of time in practicing parades.

  Let us look at another example: The duty of an employee is for example i) to produce output and ii) to care for the expensive machines he uses. While the output can be easily counted the maintenance of the machine is harder to observe. If the employee wants to maximize his payroll he will put all his effort into the maximization of output. Until the expensive machine has a malfunction because of limited maintenance it may take years and the employee may be confident that he is retired at that time.

  Similarly, individuals will choose to focus on tasks that are easy to accomplish. In an experiment of Shapira (1976) subjects were presented seven challenging tasks that varied in their difficulty. Not astonishingly, subjects who were paid only if they succeeded have chosen easy tasks while subjects paid irrelevant of success haven chosen tasks that are more difficult.

[422] An example may be a question to inquire whether members of the parliament polish their shoes and iron their shirts themselves to find out if they are a guide to the population (Hubmann 2007).

- ***Impossibility of control:*** Control and monitoring are not only costly but sometimes also impossible.[423] Computer security expert Bruce Schneier (2006) notes that the problem of incompletion of monitoring and control can be recognized by the fact that even prisons do not succeed in keeping weapons out of their institutions. If it is practically not possible to keep weapons out of prisons it is obvious that a perfect control system is neither realistic in other cases. A bystander will have a hard time determining by observation the marginal productivity of two men jointly lift heavy cargo into a truck (Alchian & Demsetz 1972: 779).

  In the famous example of Titmuss (1970: 142-157, see also below) about blood donation one argument is that the risk of infected blood is much greater for paid than for volunteer blood donors. The quality of donated blood from individuals motivated out of a sense of duty is better then the one given from paid individuals. By 1971 this relationship between paid donors and increased risks has been accepted by most experts (Domen 1995: 53). In fact, experience has shown that the risk of a disease transmission is 10 to 20 times greater with paid donors than with volunteer donors (Eastlund 1998; reported in Eastlund 2003: 1513). Similarly, an analysis of different 28 studies concluded that paid donor populations have higher frequencies of blood borne infections than unpaid donors (Van der Poel, Siefried, & Schaasberg 2002: 291).

  The problem is that blood quality cannot totally be controlled. The window period for a test is in medicine the time between infection and the moment after which it is possible to detect it. In this time it is not possible to control whether somebody has a certain disease. There are no screening tests without a window period (Van der Poel, Siefried, & Schaasberg 2002: 292). For HIV the average window period with antibody tests is 22 days, with antigen tests 16 days, and with nucleic acid test 12 days (FDA 2001). For hepatitis C the average window period can maximally be reduced to 25 days. In this window period the only possibility to prevent infected blood to enter blood banks seems to be that risk donors refrain from giving blood or anonymously state belonging to a risk group.

  If a donor belongs to a risk group it depends on his own willingness (Arrow 1972: 345)[424] respectively a sense of duty that he refrains from donation. If donors are motivated by receiving a monetary reward refraining from donation incurs a monetary loss.[425] Individuals in need of money probably donate blood even if they belong to a risk group. The blood donor center in Zurich (Switzerland) states that volunteerism is one of the most important safety factors in blood donations (Tommer 2006: 43). That is probably why in many countries it is by law forbidden to pay for blood donations as in Austria.[426/427] A study on payment for blood donations found that out of twelve nations only in the

[423] For a short review on literature on control systems at the workplace see Sackett and DeVore (2001: 159-160).

[424] In the early 70ies there was no test to detect hepatitis in the blood. However, it may be assumed that Arrow's argument also holds true for the window period.

[425] A solution is to pay for not donating if somebody belongs to a risk group. In this case the donators have no incentive to lie about belonging to a risk group. However, doing so does almost for sure create huge problems.

[426] The Blutsicherheitsgesetz (Republik Österreich 1999, §8(4)) states the following: "Es ist untersagt, Spendern von Blut oder Blutbestandteilen oder dritten Personen für eine Spende einen Gewinn zukommen zu lassen oder zu versprechen" (own translation: It is forbidden to give or promise a benefit to donators of blood or blood elements).

[427] Domen (1995: 55-57) argues in a historical review of paid blood donations and related risks that emotions in the population and moral concerns played an important role in the ban of paid donors. While emotions and moral concerns have likely played a role the claimed influence seems subjectively to be exaggerated. Unfortunately, Domen does not provide any data in support of his claims.

United States[428] and in Germany[429] donators are partially paid (Engelfriet et al. 2006: 63). In the United States the FDA (2006) requires that all paid donations[430] have a "paid donor" label attached on blood bags.

The example of blood donation presented above is not a singular case. The willingness of being honest is often required in economic life (Arrow 1972: 345-346). In many exchange situations truth, trust, loyalty, and justice are required. Some characteristics of human behavior are hardly controllable.

- ***Undermining intrinsic motivation:*** A problem is that externally implemented incentives may undermine intrinsic motivation.[431/432] This has already been acknowledged by Woodworth (1918: 69-70). Aristotle (1998: 259) argued similarly that the pleasure from the outside might destroy the pleasure coming from the activity itself. It may therefore be that such external inducements may have counter-intentional consequences and do not contribute to solve social dilemmas (Ostrom 2000b: 5). Psychological research has shown that intrinsic motivation is undermined by external intervention if individuals feel that their self-determination is adversely affected respectively if they consider these interventions to be controlling (Deci, Koestner, & Ryan 1999).

  Sanctioning may also have the consequence that the judgment of appropriateness (March 1994) is changed respectively the perception of what type of behavior is suitable, what the expectations of others are, and what norms are applicable (Tenbrunsel & Messick 1999: 687). Before acting individuals ask themselves unconsciously: “What kind of situation is this”? If they consider the situation as competitive they start competing and if they consider the situation as cooperative they start cooperating (see also Wall Street vs. community game in chapter 5.4.2.2). Such consequences may arise if at least some individuals are not rational egoists (Sen 1977: 330)[433] and follow norms such as for example the norms of reciprocity and fairness. Sanctioning does therefore not only change the payoff but also the perception of the situation. Ethical and moral considerations are replaced by norms of competition. If an individual does not defect because he is pro-socially motivated introducing a fine may prompt him to stat thinking about the situation from an individual cost-benefit point of view (Messick 2000: 235).

---

[428] However, the number of paid donors has strongly diminished and nearly all blood donations are collected from volunteers (Engelfriet et al. 2006: 63).

[429] In Germany there are three different donation systems. Red Cross donors are not paid, University and Community Blood Services usually pay their donors, and commercial plasmapheresis centers generally pay their donors (Engelfriet et al. 2006: 63 and 65-66). These payments have to be in accordance with the Transfusionsgesetz (Bundesrepublik Deutschland 2006, §10(1)) that allows only payments to cover the expenses of the donors. There has been a debate in Germany for several years whether blood donors should be paid or not.

[430] This includes incentives such as event tickets, music compact discs, reduced hotel room rates, medical tests, or scholarship programs (FDA 2006). The label “paid donor” has to be attached “whether or not the incentive is offered only to donors who are successful in donating or if all donors who present to donate receive the incentive”.

[431] A very interesting story related to the undermining of intrinsic motivation is reported by HP founder David Packard (1996: 135-136). While General Electric looked up their expensive tools so that they were not stolen HP determined that the tools storeroom should always be open. As a result, General Electric faced the problem that employees stole tools whenever they could. For HP the open door policy was a symbol of trust that in turn was honored by the employees.

[432] For a very critical meta-analysis of the undermining effect see Cameron et al. (2001; see also 2002). In fact, they found “no evidence for detrimental effects of reward on measures of intrinsic motivation” (Cameron, Banko, & Pierce 2001: 21).

[433] Sen (1977: 330 and 333) argues that commitment, which is similar to pro-social behavior, is not important for consumer behavior but for collective action and work motivation.

Structural solutions for solving the problem of collective action are only successful if the positive incentives or the punishments are stronger than the undermining of intrinsic motivation (Weibel, Rost, & Osterloh 2007).[434] Otherwise, punishment and positive incentives do not lead to the desired behavior. In fact, a series of experiments by Tenbrunsel and Messick (1999) showed that sanctions might destroy expectations about other's cooperation. If the sanctions were not high enough individuals stopped cooperating making the result worse than before the sanction were introduced. Weak sanctions intended to increase cooperation may in fact reduce it (Tenbrunsel & Messick 1999: 704).

Another negative by-effect of a structural solution is that scare resources are used to extrinsically motivate individuals who contribute anyway out of intrinsic motivation. One may therefore conclude that there are not only hidden costs of control (Falk & Kosfeld 2006) but also hidden costs of structural solutions in social dilemmas.

The probably best-known example for an undermining effect is reported by Titmuss (1970: 239) who argues that monetary compensation for donating blood might reduce the supply of blood donors.[435/436] By paying some donors the nature of transaction is changed and the satisfaction of volunteers being indispensable may be destroyed (Williamson 1975: 38). In fact, a field experiment showed that blood donations decline if monetary compensation is offered (Mellström & Johannesson 2005: 9-10).

That the enjoyment of searching and seeing may be completely destroyed by external coercion has been confirmed by Einstein (1949: 17-18). He stated that he believes that it is possible to rob even a healthy beast of prey of its voraciousness if it is forced with the aid of a whip to devour continuously.

A review of economical and psychological literature concluded that there is strong empirical evidence for undermining effects and reinforcement of intrinsic motivation (Frey & Jegen 2001: 606).[437] However, an undermining of intrinsic motivation does not always happen (Amabile 1997: 45-46). External interventions undermine intrinsic motivation only if it is perceived as controlling. Further, the initial level of intrinsic motivation plays a crucial role. An undermining effect is more likely if the intrinsic motivation is weak, vague, or ambiguous. The timing of extrinsic motivation is also important. Extrinsic motivation may be helpful for stages in which the task does not depend on novelty such as the gathering of background information or the validation of a chosen solution.

---

434 There is a discussion whether governmental social welfare crowd out private charitable giving (Nyborg & Rege 2003). Abrams and Schmitz (1984; 1985) claim that there are empirical findings that governmental transfers crowd out charitable donations. This is questioned by Driessen (1985) who claims that there are some empirical and theoretical problems in the research of Abrams and Schmitz. Schiff (1985) concludes that certain types of government spending such as cash transfer to the needy decrease private charity while other types such as spending on social service increase it. According to Nyborg and Rege (2003: 399) the majority of empirical studies indicate that governmental contributions crowd out private contributions but that the crowding-out is incomplete. In an interesting large-scale study on voluntarism in Canada the authors researched the influence of governmental spending on spent time and not on donated money as in other studies (Day & Devlin 1996). They found that governmental spending affects the decision to volunteer negatively but if somebody volunteers the number of hours donated is not affected.

435 The argumentation of Titmuss appears to be somewhat ideologically biased. For reflections of his arguments see Arrow (1972), Kessel (1974), and Solow (1971); for a short comment see Williamson (1975: 37-38).

436 For an expanded and updated edition see Titmuss (1997).

437 Frey (2001) uses the terms crowding-out and crowding-in. To avoid a confusion of these terms with their use in the discussion whether debt-financed deficits "crowd-out" interest-sensitive private sector spending (Friedman 1978: 596) they are not used in this dissertation (see also footnote 434).

- ***The downside of coercion itself:*** Empirical evidence indicates that those societies where people are best off are those whose government corresponds least to the authoritarian Leviathan (Orbell & Rutherfo 1973). The analysis of Orbell and Rutherford (1973: 403) showed that a Leviathan does not bring social order and a civilized life.

  Even if coercion or sanctions were effective the violation of individual rights may not be accepted. While a sanctioning system may promote collective efficiency it deprives individuals of personal autonomy and freedom (Mulder et al. 2005: 444-446). Based on a series of experiments Baron and Jurney (1993) come to the conclusion that in a social dilemma situation the preference for maximizing the efficient use of a good may be less important than the maximization of other preferences (see also DeTienne 1994). There is certainly a maximum level of coercion and monitoring under that one feels still agreeable. Probably few of us wish an Orwellian (Orwell 1949) type of a Leviathan who is constantly watching us even if thereby a collective good could be perfectly provided. However, if "things went wrong" and the inefficiency is too big individuals accept the downside of sanctions and strong leadership on autonomy and personal freedom (Mulder et al. 2005: 453; Samuelson & Messick 1986: 961; Van Vugt & de Cremer 1999: 595).

- ***Cooperation stability:*** Based on simulations Kondo (1990) proposed that mutual cooperation or social order might be maintained by farsighted and perfectly rational individuals who expect reciprocity. However, he argues that this cooperation is not stable. Disturbances or deviant behavior destroys cooperation very fast. Social order in a dictatorship collapses very fast if the resources for punishment get inefficient. The rapid collapse of the German Democratic Republic is a corresponding example. At a certain point the government lost the ability to punish participants of the Monday demonstrations (see chapter 6.4).

  Normative behavior produces more stable cooperation even under disturbances and works as a stabilizer. If individuals act not only normatively but even morally they act as catalyzers that can convert non-cooperation into cooperation.

5.6.2 Motivational solutions are not without pitfalls

Rational choice analyses are often criticized of being *undersocialized* (Granovetter 1985: 483-487). Indeed, as argued above solving social dilemmas only with reference to a "nonsatiety of economic greed" seems to be not always beneficial (Dawes 1980: 170). However, there are also *oversocialized* concepts of man (Wrong 1961). Concepts that assume that human beings have no other chance than being thoroughly socialized are a misinterpretation of reality. Assuming that it needs pro-social motivation to solve social dilemmas leads to the question how to get everybody to have such preferences. Indeed, the motivational solution is obviously not without its problems:

- ***Short supply of intrinsically interesting tasks:*** It seems to be reasonable to assume that there are jobs and tasks that are intrinsically more interesting than others. While many physicians are probably motivated by the task of saving lives few physicians love the related paperwork.[438] It is also hard to

[438] The recommendation of a physician is very important for the decision of a subject to participate in a clinical trial on cancer treatment (Avis et al. 2006: 1867). At the same time the referral of a physician to such trials depends significantly on the considered amount of involved paperwork (Siminoff et al. 2000: 1210). If the corresponding

imagine that there are many individuals intrinsically motivated making cold sales calls (Lawler 2000: 68).[439] Certainly, there are many tasks that are intrinsically rewarding such performing in a play or making pottery. However, it seems to be clear that a society that focuses only on such tasks will not succeed. It can be therefore argued that there is short supply of tasks that are intrinsically satisfying *and* necessary for societal survival. In consequence, jobs that are intrinsically less rewarding have to be promoted by extrinsic rewards.

- ***Short supply of intrinsically motivated individuals:*** Not all individuals are to the same extent intrinsically motivated. While probably some stock traders are to some degree motivated by the fun of trading others are perhaps more motivated by their provision. Only hiring intrinsically motivated employees may work for special organizations such as small architecture firms, small software companies, or small advertising agencies. However, this will not work for many other organizations because highly intrinsic motivated employees are in short supply. A regular firm may have trouble finding only intrinsically motivated employees.

  Certainly, the above described problem gets smaller in the age of the Internet. It is easier to find an individual doing x or y out of intrinsic interest globally than in the neighborhood. In a narrow neighborhood there are probably not many individuals whose hobby is the geology of the Mars. However, all over the world there probably quite a few enthusiasts interested in this topic. That is probably why the NASA Clickworkers experiment in which volunteers classified Mars craters *over the Internet* was successful.

- ***Matching organizational needs and individual interests:*** Even if an individual is highly intrinsically motivated and the task is potentially intrinsically interesting there is the problem whether the motivation of the individual and the task do match. A devoted scholar may love to do a mathematical proof because of its beauty but it may be a pain for a college student to do so. At the same time the student possibly loves snowboarding while this is perhaps no fun for the scholar. In this situation, to organize the workload efficiently the scholar has to be assigned to do math while the student has to be appointed as a snowboard instructor. Such a matching of organizational needs and individual interests is on the one hand not trivial and on the other hand costly.

- ***Finding intrinsically motivated individuals:*** It may be hard to know the extent of intrinsic or extrinsic motivation of a job applicant. Most job applicants will write in their motivation letter that working on x or y is by coincidence exactly what he or she has ever dreamed off. Due to self-selection mechanisms it may be that for some areas such as religious orders only intrinsically motivated individuals or only honest individuals apply.[440] However, for most organizations this will not be the case. Figuring out the extent of intrinsic motivation is therefore not trivial.

- ***Difficult to manage:*** For extrinsic motivation the relationship between motivation and incentives is clear: the more the better. If it is thought that an employee is not motivated enough the provision has

paperwork is considered too time consuming the likelihood of a referral to clinical trials decreases. For an overview over reasons for resistance of physicians towards clinical trials see Castel et al. (2006: 1746-1747).

[439] For a description of how it is to be a cold caller see Ronson (2006).

[440] In the medieval age joining a religious order could also have other incentives than religion. Elm and Feige (1980: 238; see also Kieser 1987) report that in some cases aristocrats got abbot of a monastery solely to make as much money as possible.

simply to be increased. For intrinsic motivation one has to find out individually how each employee is motivated. Furthermore, the relationship between motivation and incentive is quite complex. Too much stimulation may be as bad as too little. Extrinsic motivation is therefore certainly easier to manage than intrinsic motivation.

- ***Intellectually stimulating but not demanded:*** Intrinsically motivated solutions to social dilemmas may be in some cases undesirable (Frey & Osterloh 2002: 22; Osterloh & Frey 2000: 540; Osterloh, Frey, & Frost 2001: 234). What is fun is not always best for organizational effectiveness (Lawler 2000: 68). An architectural solution that is the most creative and the most intellectually stimulating is not always (but certainly sometimes) the one that wins a design competition. Because architectural firms have to win such competitions from time to time to pay their expenditures and making a profit intrinsic motivation may not always be welcomed.
- ***Costs of intrinsic motivation:*** Motivating people intrinsically may be costly. One has to invest scare resources such as time and money to keep employees fascinated. This may involve letting an employee work on a project his fascinated in but that is financially unsuccessful. At Google there is for engineers a "20 percent time" rule (Google 2007). It states that engineers are in 20% of their work time free to pursue whatever they are interested in. While such a rule is probably good for intrinsic motivation[441] it is also very costly. That is probably the reason why the "20 percent time" i) does only apply to technical staff but not to sales staff or the management (Battelle & Schmidt 2005) and why it is ii) limited to 20% of the work time.
- ***Honored more in the breach than in the observance:***[442] Subjects present themselves usually more favorable with respect to norms and standards. Similarly, it is likely that cooperative behavior is in some cases announced in public but not kept in private. This does also apply to pro-social behavior, intrinsic motivation, and norms. In empirical research the phenomenon of socially desirable responding is quite known (e.g. Arnold & Feldman 1981; Fisher 1993; Fisher & Katz 2000; Zerbe & Paulhus 1987).

5.6.3 Structural and motivational solutions are complementary

It is not argued here that the problems associated with motivational solutions for social dilemmas are smaller than the ones associated with structural solutions and vice versa. Rather the different mechanisms to facilitate the production of collective goods seem to have its own advantages and disadvantages. They are in consequence complementary (Elster 1989b: 147). The police as a governmental institution may fine me if I litter in the park. This is probably a good solution to prevent littering of noncompliant individuals. However, the police have to be paid; that is why police presence must be limited. Nevertheless, even if there is no

[441] This rule is probably why many high-rated FOSS hackers joined Google. They are paid for working on their FOSS project without the fear that their employer interferes. Google's "20 percent time" rule may not only serve to foster intrinsic motivation or attract employees but also by boosting innovation. According to Vice President of Search Product and User Experience Marissa Ann Mayer (2006) half of the new product launches in 2005 originated from such "20 percent time" projects.

[442] The subtitle refers to a conversation between Hamlet, Horatio, and Marcellus. Hamlet states in reference to a custom that it is "more honored in the breach than in the observance" (Shakespeare 1992a: 49). In the play "Hamlet" it refers to a bad custom that should be violated and which is only thereby honored (Jenkins 2007). This interpretation of the phrase is clearly different from the one used here. However, it seems that the phrase is usually used not in the original sense as it is done here (e.g. see Coffee 2001: 2171; Miller 1936: 86).

policeman around I probably prefer not to litter the park because I do not want to damage my good reputation. It also may be that I do not want to expose myself to the social pressure if people glare at me. Such mechanisms do not work in the evening where it is hard to recognize individuals or in remote parts of the park. Yet, many individuals do not litter although nobody is around and they know for sure that they will get away unharmed. There may be an inner conviction that one should not litter the park. This short example shows that there are many different types of motivation why people do not litter a park. Each motivation works better in some situations than in others and each motivation has advantages and disadvantages. Fining for examples only works if the police is around. At the same time not all individuals share the inner conviction that littering is a bad thing to do.

Elster (1989b: 131) suggests that the problem of collective action is solved with a two-step process: individuals motivated by altruism or pro-social behavior set up a central institution with the capacity to enforce cooperative behavior. Afterwards rational egoists cooperate as well because of the payoff structure maintained by the institution. In this view it is then wrong “to believe that each instance of cooperation can be explained by one motivation” (Elster 1989b: 131). Based on this argument it is thought here that differently motivated individuals are complementary in solving social dilemmas (see also chapter 6.3).

5.7 Production function of collective goods

A production function of a good expresses the relationship between input and output. This holds also true for collective goods. Furthermore, the production function is a very important characteristic of a collective good. Oliver and Marwell (1985: 523) state that “different types of production functions create dramatically different dynamics in otherwise similar situations and thus lead to different outcomes”.

Continuous versus step level public goods

There are two generic models of production functions of public goods (Eek, Biel, & Garling 1999: 2; Komorita & Parks 1995: 193): continuous, linear (see F = 1.0 in Figure 69a) and step-level (see Figure 69b) public goods.[443]

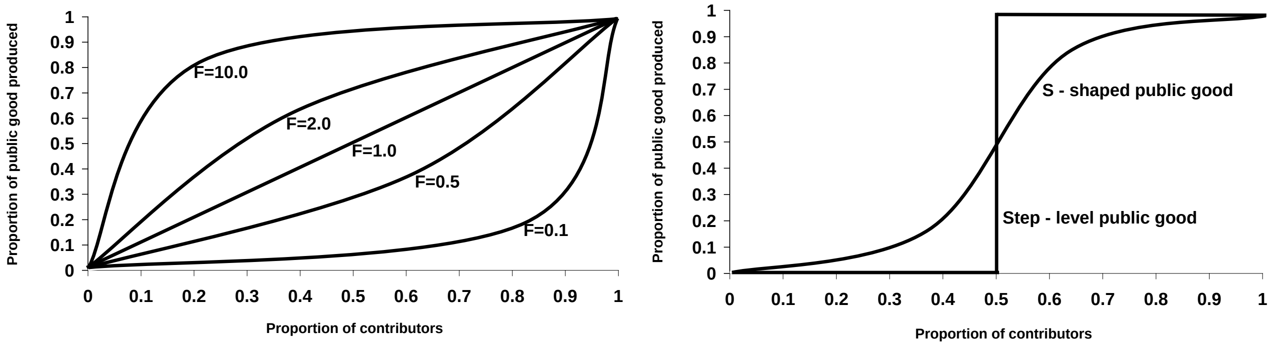


Figure 69ab: Production functions of public goods[444]

[443] For Kollock (1998: 189-190) there are four basic production functions: linear, discontinuous or step-level, decelerating, and accelerating production functions. In a classification of Marwell and Oliver (Marwell & Oliver 1993: 58) additionally a third order (S-shape) production function is thought to be basic.

Most studies on public goods presume a linear production function in which the value of the public good varies directly with the cooperation level. Collective outcome increases for continuous public goods with the number of cooperators.[445] The more people contribute, the higher the value of the public good. An example is public broadcasting radio in the United States that is funded by voluntary consumer donations. The more money is donated the more types of radio stations can be funded.

For step-level public goods there is a provision point. The public good is provided if and only if the total amount contributed exceeds the provision point (Sonnemans, Schram, & Offerman 1998: 144). In this model the value of a contribution varies heavily during the production process. The contributions donated before and after the provision point is reached do not change the value of the public good. An example is the lighthouse.[446] The first bricks will not do any good and after the lighthouse is working any further contribution will not change its performance substantially. The provision point for a lighthouse is reached if the light bulb is inserted. Without this light bulb the lighthouse is useless while any further contribution will not change lighthouse performance drastically. Another example is voting. The most important individual is the one whose vote turns the minority into a majority (Marwell & Oliver 1993: 60). Every vote below or above this point is not completely but almost useless. That is why in non-proportional voting systems with multiple districts and single winners gerrymandering is practiced.[447]

A major characteristic of a step-level public good is that no unconditionally best strategy can be identified (Rapoport 1985: 149). For step level goods at some level of collective contribution "rational individuals may conclude that is worthwhile to contribute toward the supply of the collective good" (Frohlich, Oppenheimer, & Young 1971: 21-22). This is the case if one expects "that others' contributions would fall short of the cost of providing a step good" (Hardin 1982: 58). In the lighthouse example it may be rational to provide the light bulb to complete construction and enable thereby proper functioning of the lighthouse.

Next to step-level and continuous public goods there are public goods with production functions between these two extremes (Oliver 1980: 254-255). Continuous public goods may benefit slightly more from the very first or the very last contributions. In Figure 69a F is an exponent controlling the shape of the production function (Heckathorn 1996: 255).[448] If F is bigger than 1 the production function is decelerating. The first contributors are the most productive while succeeding contributors face decreasing marginal returns. There is therefore a negative interdependence between contributions because each one makes subsequent contributions less worthwhile (Marwell & Oliver 1993: 61). An example may be entertainment at a family reunion. The first contribution will cheer up the crowd while the tenth contribution will hardly be appreciated anymore. For a production function with F = 2.0 the first 10% of contributors produce 19% of the public good while the last 10% produce only 1% of the public good. However, if F is smaller than 1 the production function is accelerating. For F = 0.5 the first third of contributors produce only 12% while the last 10% of contributors produce more than one third of the public good. For such an accelerating production function there are

[444] Adapted from Heckathorn (1996: 252).

[445] Certainly, if subjects contribute different amounts the contribution level does not completely increase with the number of cooperators.

[446] For a discussion why the lighthouse, despite the reasoning of Coase, is a good example of a public good see chapter 5.3.

[447] Gerrymandering is the practice of changing electoral districts boundaries so that gained seats are maximized (Anonymous 2006; Cox & Katz 2007). To do so voters from districts with clear victory are assigned to hotly contested districts. Thereby no votes are "lost".

[448] $L = 1 - (D/N)^F$, whereas L is the level of the collective good produced, N the number of involved actors, and D the number of defectors (Oliver 1980: 255).

positive interdependencies (Marwell & Oliver 1993: 63). Each contribution makes the next contribution more worthwhile. Typically, mass actions such as strikes and boycotts, for which success is based on the number of participants, have such a production function.

The production function of common goods

Production functions are also important for common goods. Some common goods degenerate linearly with overuse and defective individual choices. An example is spamming that overuses bandwidth linearly. Two junk mails forces us to delete two mails, four junk mails force us to delete four mails. Other common goods have the power to compensate for some degree of overuse until they collapse completely at a certain point, like for example overfishing. Non-cooperative behaviour of a few fishing boats do not lead to irreversible overfishing (Myers & Worm 2003; for a critical comment on the Myers/Worm paper see Polacheck 2006). However, if too many fishing boats do not cooperate, fish population will collapse immediately and at once because fish need a certain community size for successful reproduction (Zabel et al. 2003: 152-155).[449]

The production function as a deliberate choice

Collective goods differ from one another in many ways (Marwell & Oliver 1993: 23-24). Office microwaves, air pollution, or national defense are quite different from each other. Some of these differences are reflected in the production function discussed above (Marwell & Oliver 1993: 24). However, most collective goods can be produced in more than one way (Marwell & Oliver 1993: 28). The production function of a collective good is sometimes the result of a deliberate choice. An example is perhaps code reuse and the concept of modularity in software development. To develop a certain functionality without reusing code one has to invest a lot of resources because the software has to be written from scratch. The first contributions will not do any good until a critical mass is reached and the software is running. However, if source code is reused or a module is added to an existing software project already the first contributions will make a difference.

The consequences of the production function

The shape of the production function of collective goods has a big influence on its production. Although the basic dilemma remains the same collective goods with different production functions face different problems:

- For a production function with increasing marginal returns a nearly universal contribution is required to produce a meaningful amount of a public good. The start-up problem gets prominent and collective action does not get easily off the ground (Marwell & Oliver 1993: 92). However, if this problem is solved additional contributions are more beneficial and therefore more likely. After a certain point contribution may get almost self-enforcing.

[449] Additionally, the collapse of one fish species may lead to the collapse of other fish species not directly affected by fishing. The reason for this is that certain species need a certain fish community composition. There are complex species interactions in ecological systems. Environmental perturbations may or may not have an influence on ecosystem depending on whether a dominant or keystone species is affected or not (Brown et al. 2001).

- A production function with decreasing marginal returns faces only a minor start-up problem. At the same time with every extra contribution the production of a public good gets more and more difficult. A finishing problem emerges.

Marwell and Oliver (1993: 60-61) discuss public goods that are characterized by a third order production function (see Figure 70).

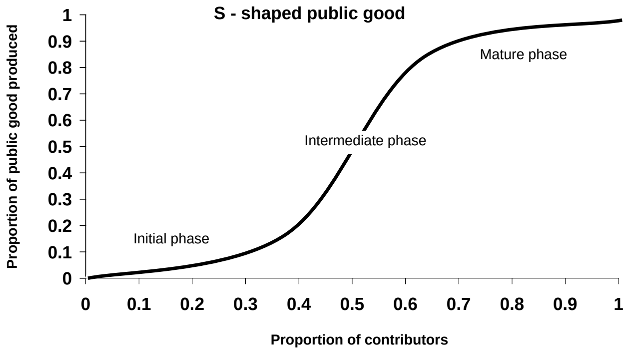


Figure 70: S-shaped production function of a public good

Such an S-shaped public good approximates a step-level public good (Marwell & Oliver 1993: 60). At the beginning of production there are low but increasing marginal returns to initial contributions (Heckathorn 1996: 252). The very first contribution will not do much good. Because the return to such contributions is very low a start-up problem emerges. Strongly motivated contributors must be willing to bear the starting costs. At this point of the production process free riding is not a problem because there is nothing to free ride on (Marwell & Oliver 1993: 182). The problem is rather to get individuals to contribute. If a critical mass of individuals has contributed the production function becomes linear (Heckathorn 1996: 252). The relation between input and output changes in the favor of the later. In this phase even rational egoists may consider it beneficial to contribute because they receive a net gain from contribution. At last, marginal returns decrease, output limes are approached, the curve of the production function decelerates, and a finishing problem emerges. Heckathorn (1996: 272-276) recognizes therefore three phases.

i. In the ***initial phase*** there is a start-up problem and an altruist dilemma. It needs ***zealots***[450] to reach a critical mass.

ii. In the ***intermediate phase*** the marginal returns for contributors become greater. ***Strategic interaction*** or ***selective incentives*** may advance the production of the public good.

iii. In the ***mature phase*** the limits of the production of the public goods are approached. Overproduction may be a problem.

Based on this argumentation solving social dilemmas is dependent on the production phase of a public good. In an initial phase it needs strongly motivated individuals while in an intermediate phase strategic interaction

[450] The Zealots were a Jewish resistance movement in the first century who rebelled against the Roman Empire (Horsley 1986; Kaufmann 2002). By violent actions the religiously motivated Zealots tried to provoke a Roman overreaction respectively brutal repression to stimulate a broader Jewish revolt that “the terrorists could never have stimulated directly” (Hess 2003: 340). As planned by the Zealots the Romans answered with bloody repression. As a consequence, the Jewish people went to war against the Romans.

or selective incentives may be the solution. This also highlights the argument that social dilemmas are solved incrementally (Heckathorn 1996: 272, see also chapter 6).

5.8 Social dilemmas in real life business

Weber et al. (2004: 281) argue that social dilemmas are everywhere and that it is difficult to imagine a sphere of social life that is not affected by it. Similarly, Arrow (1974: 26) concluded that commercial firms, political parties, revolutionary movements, churches, and all other organizations "share the common characteristics of the need for collective action and the allocation of resources through non-market methods". Indeed, it seems that it is "the essence of an organization that it provides an inseparable, generalized benefit" and that in consequence "the provision of public or collective goods is the fundamental function of organizations generally" (Olson 1995: 15). Deduced from this observation it seems to be clear that social dilemmas are also an integral part of real life business. According to Alchian and Demsetz (1972: 777) work within a company as well as trade in markets need cooperation. They further argue that it is an illusion that within firms issues can be solely settled by fiat and authority. The difference between market transactions and transactions within a firm is that in the later teamwork is involved (Alchian & Demsetz 1972: 778) which is especially prone to be subject of social dilemmas. It can be concluded that "the strategic difficulty faced by the firm and the worker is an instance of the general problem of cooperation" (Posner 2000: 13).

Social dilemmas in firms

At the department level of big firms it is best for each department to consume as much resources such as budget, employees, or space as possible (Tenbrunsel & Messick 1999: 686). However, collectively this behavior reduces the viability of the firm and hence of its individual departments. Similarly, at the employee level it is individually rational to shirk but if all employees adopt this strategy the firm suffers and every individual employee is worse off than if each had expended some effort. Examples may be quality circles in which members are supposed to share their ideas for improving quality and where there are group-oriented reward systems (Sniezek, May, & Sawyer 1990: 155-156).

Schneider and Northcraft (1999) describe three social dilemmas in organizations related to workforce diversity. *Firstly*, there is a dilemma of the organization. Although it is in the interest of society that organizations support diversity it may be costly for the individual organization to do so and therefore it is better if other organizations assume this responsibility.[451] *Secondly*, there is a managerial dilemma. Though it may be in the interest of the organization that managers promote diverse work groups, it may be in the interest of a particular manager that other managers promote this responsibility. *Thirdly*, there is an individual dilemma. Even if it is in the interest of an organization or a manager to further the pursuit of diversity it may be in the interest of an individual to further the own goals.

[451] A similar scenario for the agricultural business was constructed by McDaniel and Sistrunk (1991: 26-28).

The problem of social loafing

In social psychology literature on groups a special version of social dilemmas is known as social loafing (Williams, Jackson, & Karau 1995).[452] In teamwork social loafing is the "tendency for individuals to expend less effort when working collectively than when working individually" (Karau & Williams 1993: 681). Interestingly, the related problems are less problematic if team members consider the task in question as meaningful (Karau & Williams 1993: 700).

Organizational citizenship behavior

Cropanzano and Zinto (2000: 142) argue that organizational effectiveness is based on resolving the problem of employee citizenship that is a social dilemma. Individually everybody is better of to work to the line of the contract but collectively everybody is better off if employee's display a positive attitude. This is even more important in the face of failures or when other employees need support and assistance. In such cases it is the extra hour that employees spend that keeps the firm competitive. Organizational citizenship behavior was introduced by Organ (1977) and is a special type of work behavior. It is based on i) discretionary or extra-role behavior, ii) absence of direct or formal rewards, and iii) organizational functional consequences (Eastman & Pawar 2005: 27). Organizational citizenship behavior includes helping co-workers, sportsmanship in the workplace, innovative behavior, and compliance to formal and informal rules and regulations (Moon, Dyne, & Wrobel 2005: 9-10). Such behavior is either directed to specific individuals or toward the organization (Smith, Organ, & Near 1983). Important in relation to this study is that the motivation for organizational citizenship behavior is hardly self-interest (Eastman & Pawar 2005: 34) respectively that the corresponding costly behavior is not rewarded directly or formally.[453] At the same time, organizational citizenship behavior is functional for an organization. A personal sacrifice is made to promote collective interests. This characteristic is basically shared with social dilemmas (Daniels, Joireman, & Kamdar 2005: 81). It does therefore not surprise that the factors that lead individuals to engage in organizational citizenship behavior are similar to the ones that lead to cooperation in social dilemmas (Daniels, Joireman, & Kamdar 2005: 82-83, see also chapter 5.4).

The information commons

That companies solve social dilemmas with different success is shown by a study of Fulk et al. (2004) on information commons such as databases, groupware, data warehouses, project websites, intranets, and shared whiteboards (Fulk et al. 1996; see also Karau & Williams 1993; and Yuan et al. 2005). The overall benefit of such organizational information commons can be substantial for a collective such as a company, a team, or a network because information and knowledge are key resources (Fulk et al. 2004: 583). At the same time it is difficult to convince individuals to provide information which they already posses. To contribute information involves considerable costs such as time. A consultant paid by the hours solving customer problems will incur

[452] The diverse theoretical frameworks used in the social loafing literature (Karau & Williams 1993: 682-684) are somewhat different from the social dilemma literature. Despite such differences there is clearly a connection between social loafing and social dilemmas.

[453] It is empirically not always easy to distinguish between organizational citizenship behavior that is not based on self-interest and constructive utilitarian behavior that is based on self-interest. However, without going into details it seems to be plausible to assume that organizational citizenship behavior is important and that it exists.

a loss if he tries hard to contribute his solution to a database used by the whole company. Free riding in the case of information commons is very difficult to detect “since people do not have windows” through which one may see “what unshared information other people possess” (Fulk et al. 2004: 571). Without knowing what other’s know one can hardly create an effective mandatory contribution system. As Kim and Mauborgne write (1997: 71):

> *Unlike the traditional factors of production - land, labor, and capital - knowledge is a resource locked in the human mind. Creating and sharing knowledge are intangible activities that can neither be supervised nor forced out of people. They happen only when people cooperate voluntarily.*

Rewarding individuals for contributions to an information common may lead to a flood of low quality information (Fulk et al. 2004: 571; Kalman et al. 2002: 126). However, extrinsic motivation may also work fine. In two out of three surveyed companies contributions of individuals to the information common were positively related to the individual gains (Fulk et al. 2004: 580).

5.9 Summing up

The basic insights of this chapter are:

- It is inevitable that individuals of a society share common goods. In consequence, social dilemmas arise in which each individual receives a higher payoff for a socially defecting choice than for a socially cooperative choice and all individuals are better off if all cooperate than if all defect. > See chapter 5.1
- Public goods are characterized by non-rivalry and non-excludability. In some cases there is not only non-rivalry in consumption but additional users may furthermore increase the value of a public good. > See chapter 5.3
- There are not many goods that show pure non-excludability. In some cases the exclusion is possible but the related costs are too high. The costs of excludability rest heavily on available technology. Law respectively politics influences also whether excludability is possible. Based on deliberate choices some goods are either private or public. An example is software. Whether prospective users can be excluded or not is determined by its license terms. > See chapter 5.3
- Different social motives such as altruism, cooperation, competition, and individualisms influence the level of cooperation in social dilemmas. > See chapter 5.4.1
- Gender does not seem to influence the cooperation level in social dilemmas. > See chapter 5.4.1
- The payoff structure of a social dilemma influences the cooperation level. Not only the own payoff but also the one of others may be relevant. > See chapter 5.4.2.1.1
- Individuals cooperate more likely if they think that their contribution is critical for the provision of a collective good. > See chapter 5.4.2.1.1
- If contribution to social dilemmas is continuous instead of discrete cooperation may rise. Thereby the risk of being deceived decreases that makes an evolution of cooperation possible. In addition, contributions can be adjusted. > See chapter 5.4.2.1.1

- Whether decisions are made simultaneously or sequentially may have a huge influence on cooperation in social dilemmas. Deciding sequentially creates information asymmetries: the subsequent players know whether the first player cooperated or not. This enables conditional cooperation. Sequential decision making creates also commitment asymmetries. The first player may make a very strong commitment because he cannot revert his decision. > See chapter 5.4.2.1.1
- Having an exit option in social dilemmas may increase the overall cooperation because more cooperators than non-cooperators choose to play the game. > See chapter 5.4.2.1.1
- Fairness is an important factor influencing cooperation in social dilemmas. Aspects of procedural fairness are the possibility to voice the own opinion and involvement in decision making. An aspect of distributive fairness is the equality rule: Everybody should be treated equally. > See chapter 5.4.2.1.2
- A leader may provide selective incentives. This alters the values associated with certain actions and influences individuals' perceptions. If a leader wants to influence perceptions the success of doing so depends on his legitimacy. > See chapter 5.4.2.1.2
- Culture and the corresponding institutions influence behavior in social dilemmas. > See chapter 5.4.2.1.2
- The context of a social dilemma influences the expectations of an individual and therefore also behavior. > See chapter 5.4.2.2
- Both structural as well as motivational solutions have its own problems. > See chapter 5.6
- The production function is a very important characteristic of collective goods. If marginal returns are increasing a nearly universal contribution is required and a start-up problem emerges. If marginal returns are decreasing a finishing problem emerges. An S-shaped public good approximates a step-level public good and has a start-up and a finishing problem. To solve a start-up problem especially motivated individuals are needed. In the middle phases the relationship between input and output is getting more favorable and the problems related to collective action decrease. Strategic interaction and selective incentives may be the solution to solve the social dilemma. > See chapter 5.7
- Social dilemmas are prominent in real life business, specially for information commons. > See chapter 5.8

The preceding chapters have illustrated the object of the study, namely FOSS, motivation, and governance structures respectively mechanisms. These chapters were in essence descriptive. Distending the argument one may state that until chapter five no real problem in the production of FOSS application could have been identified. Chapter five has introduced the basic challenge: the problems related to collective action. From this point on it is clear: there are major challenges to FOSS development because public goods are involved and the problem of collective action emerges.

Next to introducing the basic problem this chapter has two major functions. *Firstly*, it clearly highlights that individual differences, the decision structure, the social structure, and perceptual factors shape collective action. This questions the results of studies that infer from experiments to a social phenomenon without taking the corresponding structures and mechanisms of the object into account. *Secondly*, this chapter shows that the number of factors influencing collective action is immense. Results of studies related to social dilemmas should therefore always be accepted with caution. One never knows whether a crucial factor influencing collective action has been overlooked.

Development of the hypotheses

Chapter six connects motivation and social dilemmas in a dynamical perspective. The importance of drive is thereby clearly outlined. It is argued that there are different phases in the production of collective goods. Very briefly ***chapter seven*** pinpoints that there are also different phases in groups and organizational life cycles. Thereby it puts the dynamical perspective discussed in chapter six in a wider organizational context. In ***chapter eight*** the hypotheses are compiled. The motivation of FOSS developers, FOSS governance structures and mechanisms, and different development phases in FOSS projects are connected to each other. This chapter integrates thus the fundamentals of the argumentation presented in chapters two to five with the dynamical perspective discussed in chapters six and seven.

6 The dynamics of collective action: The importance of drive

At the very heart of this chapter is the observation that individuals do not act in isolation but in interdependence with other individuals (Marwell & Oliver 1993: 9). Therefore, the decision situation an individual faces is unlike the one in a Crusoe economy (for an application see e.g. Ruffin 1972).[454] The decision situation is rather formed by a social exchange economy (Von Neumann & Morgenstern 1947: 10). If two or more individuals enter into exchange relations with each other the payoff for each one will not only depend on his own actions but also on those of others. In such an exchange every participant is guided by another principle and no participant can determine all variables that affect his interest (Von Neumann & Morgenstern 1947: 11). It is not possible to solve for the related behavior with probability or statistical assumptions because other people are principle guided (Von Neumann & Morgenstern 1947: 11). Individuals in a situation of social interdependence are therefore unable to optimize their own payoff independent of the choices of others (McClintock 1972: 443). Their payoff is a function of the choices of all individuals. Marshall (1910: 23) wrote that:

> *… economists … are concerned with individuals chiefly as members of the social organism. As a cathedral is something more than the stones of which it is made, as a person is something more than a series of thoughts and feelings, so the life of society is something more than the sum of the lives of its individual members. It is true that the action of the whole is made up of that of its constituent parts; and that in most economic problems the best starting-point is to be found in the motives that affect the individual, regarded not indeed as an isolated atom … but it is also true… that economics has a great and an increasing concern in motives connected with the collective ownership of property, and the collective pursuit of important aims.*

Because individuals interact with each other the importance of temporal dynamics emerges. The drivers of this dynamic are manifold. Individuals do for example not make decisions simultaneously. Some decide earlier and others later. In the following, this chapter on the dynamics of collective action is based on the following three notions:

- For certain collective goods a ***critical mass*** of contributions is needed before it is beneficial at all. The payoffs associated with contributions to collective goods are highly dependent on the amount of others who have already contributed. After a ***tipping point*** respectively a ***threshold*** is reached it may be even beneficial for rational egoists to contribute to the production of a collective good.

[454] There is a long history of economic writings (e.g. Marshall or Wicksell) that used the term Crusoe economy (White 2003). It refers to the economy of an isolated single person or an economy organized under a single will (Von Neumann & Morgenstern 1947: 9-12). In Defoe's (1719) novel Robinson Crusoe the main character lives alone on a remote island until he manages to get a servant. In his fictional book "Emile" Rousseau (1911) considers this situation as ideal for education. The only book Emile is allowed to read before the age of twelve is Robinson Crusoe. He should identify himself as Crusoe and use this situation as a "standard of comparison for all other conditions" (Rousseau 1911, Book III). This rests on the too simple assumption that a child is born good but corrupted by the social environment.

- The dynamics of collective action depends heavily on ***incremental decision making***. The decisions of individuals depend on the decision of others. A major characteristic of ***herding behavior*** and ***bandwagon effects*** is that they are based on what others have done before.
- Different individuals have ***different types of motivation***. In the production of collective goods these types of motivation are differently served depending on the development phase. That is why some individuals contribute at an early stage while others wait until the production of a collective good is almost finished. It is furthermore argued that a ***mix of motivation*** as well as ***group heterogeneity*** is very important for explaining collective action.

A methodological remark

Different models in social science often overlap (Schelling 1978: 91). This is certainly also true for the different theoretical concepts presented in this chapter. They describe a similar phenomenon: individuals do not act in isolation and the number of others having chosen a course of action matters. However, there are also differences and each concept highlights another aspect of the overall phenomenon. Critical mass describes the principle that there must be a minimal amount of actors or contributions before a single contribution makes a difference at all. The tipping point highlights that there may be a drastic change after the critical mass is reached. Schelling added a suitable diagrammatic representation of binary collective decision making. Threshold models include the insight that the amount of contributions needed to convince a single individual to contribute is different for each individual. Bandwagon effects show that the level of cooperation may gain momentum and that there may be chain reactions going on. Lock-in effects illustrate that after having succeeded to overcome the start-up problem it gets difficult to stop the bandwagon. Discussion of mixed motivation shows that the individual thresholds are partly based on the individuals’ motivation. It is furthermore argued that motivational heterogeneity may be beneficial for successful collective action.

6.1 Critical mass, threshold models, tipping points, and the like

6.1.1 Critical mass

Critical mass is the physical description for the amount of fissile material needed for starting a sustained nuclear chain reaction. However, the principle of critical mass shows also up in epidemiology, fashion, survival and extinction of species, language systems, racial integration, jaywalking, panic behavior, and political movements (Schelling 1978: 89). In such cases a threshold of participants or contributions has to be crossed before a social movement succeeds (Oliver, Marwell, & Teixeira 1985: 523).

An example of critical mass applied beyond physics is what Schelling (1978: 91-92) calls the “dying seminar” among Harvard faculty. A group of twenty-five scholars agrees to meet regularly to discuss a subject of common interest. The very first meeting has a good turnout of three-quarters. By the third or fourth meeting the attendance is about half of the interested faculty members and very soon only a handful attend the meetings. Finally, the remaining faculty members agree by consent or by an implicit understanding to cancel the seminar. What happened? By some coincidence a considerable number of faculty members do not have time to join the meeting. For some of the remaining faculty members thereby the number of individuals needed to have a proper discussion falls below the critical mass. They stop joining the meeting what causes others to do so as well. Finally, even faculty members who think that a small group can discuss the topic fruitfully will stop going to the seminar.

For Marwell and Oliver (1993: 1) the idea of critical mass is central to the understanding of collective action. Critical mass expresses the insight that it takes a minimum number of contributors or a minimal amount of contributions to convince other individuals to contribute as well. Benkler (2002: 434) observes that it is for large-scale collaboration important whether enough people can be pooled.

Related to a critical mass are step - level public goods (see chapter 5.7) which require a minimal amount of contributions until somewhat worthy at all can be produced. A lone revolutionist will not overturn the ruling order.[455] It needs a certain mass willing to overthrow government. The same relationship holds true for the building of a lighthouse. All contributions will be without any effect on the safety of sailors and fishers until finally somebody contributes a kerosene lamp respectively a light bulb. The cost for a single fisher to provide a light bulb is in contrast to the benefit of a functioning lighthouse very low. It is therefore likely that somebody will cover the costs for the light bulb. However, finding somebody who contributes the first brick of the lighthouse is more difficult because it is risky doing so. The immediate benefit is zero and there is always the possibility that the lighthouse remains unfinished.

The importance of critical mass for collective goods may be explained by two different lines of reasoning:

Firstly, the relationship between the value of collective goods and contributions is nonlinearly. Network externalities depend for example on the number of contributors and the amount of resources already contributed (Fulk et al. 2004: 570; Shapiro & Varian 1999). It needs a critical mass of telephone extensions before telephony is beneficial to most individuals.[456] The value of telephony is nonlinearly related to the number of telephone extensions.

Secondly, attribution theory argues that individuals attribute causes to events according to observations of their own and other's behavior (Heider 1958, see also informational cascades in chapter 6.2.3). Yuan et al. (2005: 219) argue that for information repositories:

> *… people's observations of contribution behaviors by other team members wile cue inferences about others team members' perceptions of the value of information repositories. These attributed value perceptions will, in turn, influence focal individuals' cognitions. Thus, when others fail to contribute or retrieve information, focal individuals are likely to conclude that others do not value a repository. This conclusion is likely to dampen individuals' own values for the repository.*

There must be a minimal amount of individuals joining collective action respectively a critical mass until an individual considers doing so as well valuable. Therefore, one may argue that not only cost-benefit analysis explains the importance of critical mass but also cognitive effects.

---

[455] At the same time this lone revolutionist is the one who convinces others to join the cause. Individuals who are highly interested play a very important role in providing the impetus for collective action (Marwell & Oliver 1993: 10).

[456] The telephony example also highlights that different individuals have different threshold values. For the President of the United States the red phone is very valuable although he can only talk with "one" person (see also footnote 140). For another person it is enough to speak with a couple of friends while for a direct marketing firm it may be crucial to reach as many potential customers as possible.

Critical mass and low-cost situations

Kirchgässner (1998: 4-5) distinguishes between four ideal types of individuals who may act altruistically:

i. The ***morally average person*** who knows that everybody should contribute to collective goods and probably does so accordingly.

ii. The ***hero*** who produces a collective good mostly on its own and bears considerable costs for doing so.

iii. The ***idealist*** who knows that many individuals should contribute to a collective good but that hardly anybody does so. Nevertheless, he contributes his share.

iv. The ***fanatic*** who knows that the collective good will never ever produced but contributes to it nonetheless.

Kirchgässner (1998: 6-7) argues that for a ***morally average person*** a necessary condition for acting morally is that the costs for doing so must be low.[457] That is why such individuals do not start cooperation. Being the first to contribute is in most cases very costly because a critical mass has not yet been reached. This does not hold true for ***heroes***, ***idealists***, and ***fanatics***. For these individuals the costs do not have to be necessarily low to secure their contribution. In consequence, it can be concluded that either heroes, idealists, fanatics, and the like form a critical mass for their own or the costs of contribution must be very low if a collective should be produced. If there are not enough heroes, idealists, and fanatics and if the costs are high the morally average person will not contribute and the critical mass will not be reached.

In most situations there are not enough heroes, idealists, and fanatics. Most altruistic individuals will be morally average persons (see also chapter 5.4.1). It can therefore be concluded that for most situations the costs of contribution must be low so that a critical mass is reached.

Critical mass and the PledgeBank example

Bagnoli and McKee (1991: 351-352) report different real world examples in which a critical mass of contributions was necessary to provide a public good successfully. In these examples it was promised that the contributions were to be returned if the contributions would fall below the critical mass. This has an important implication: Contributors may be sure that their donation is not wasted. This may happen if the provision point of a step-level public good is not reached.

An example of an organization that deliberately uses the mechanism described above to produce collective goods is PledgeBank. PledgeBank is a website that runs pledges for different purposes of the form "I will do something, *if* a certain number of people will help me do it" (PledgeBank 2006). Other individuals can subscribe to the pledge that raises the count. The promise to take action is only due if enough others singed the promise as well. This has the advantage that the subscriber gets a sort of "insurance against being the only person to show up to the demo in the freezing rain or against being the parent who discovers they've just volunteered to run the entire school play on their own." One only has to provide scare resources if a critical mass is reached and the collective good is provided. This ensures that one neither is a jerk nor spends scare resources for a desperate cause. Therefore, PledgeBank is not only an example for active management of

[457] The other necessary conditions are self-efficacy discussed in chapter 5.4.2.1.1, non-distribution constraint discussed in chapter 2.6.2.6, the morality of the object in need for contribution, and no undermining of intrinsic motivation discussed in chapter 5.6 (Kirchgässner 1998: 6-10).

critical mass but also for conditional cooperation. Many individuals agree to contribute to collective goods as long as others do as well.

There is no legal guarantee that people carry out their pledges. However, a study of several successful money based pledges found that typically three quarters of people pay what they have promised (PledgeBank 2006).

A real example of a pledge at PledgeBank is the promise of British Prime Minister Tony Blair (2006) to become the patron of a London community sports club until the Olympics in 2012 as long as 100 other public figures in London will join him. This pledge was successful and 109 public figures signed the promise. Another example targets the recognized project of several faculty members of the MIT Media Lab and corporate members like Google and AMD to provide every child in the world with a laptop which costs below 100$ (Stecklow 2005).[458] Mike Liveright (2006) promised to purchase the $100 laptop at $300 but only if 100,000 others do as well. This would in fact have resulted in the possibility to distribute about 200'000 laptops for free. However, only 3678 individuals signed the pledge and it was therefore clearly not successful.

Figure 71: Concept image of he 100$ laptop.[459]

6.1.2 Tipping points

Tipping points were first coined by Grodzins (1958) who researched American neighborhoods. He found that most white inhabitants remain in a neighborhood as long as the number of black inhabitants remained below a certain level. If this level was reached white families left the neighborhood very fast.[460] He called this point the tipping point. This idea was adapted and expanded by Thomas Schelling (1971a; 1972) and Mark Granovetter (1978).

If a critical mass of individuals is acting and a threshold near the tipping point is reached a small change may induce a dramatic cascade of other individuals acting as well (Gladwell 2000). Thereby a new equilibrium is established. Examples of how things change rapidly if the tipping point is reached are the sudden rise in sales for Hush Puppies shoes and the sudden decline of crime in the Brownsville (New York) neighborhood

[458] The machine will run Linux, X, and Gnome and allows among other things web browsing, emailing, programming, and word processing (One Laptop per Child 2006b). Technically the laptop will have a full color display, a 500MHz processor, 128MB of DRAM, at least 500MB of Flash memory, wireless network capabilities, a touchpad for mouse control and handwriting input, built-in stereo speakers and a microphone as well as 3 external USB ports (One Laptop per Child 2006a). It is intended to produce 100 million laptops. Until now Quanta Computer, the world's largest contract laptop manufacturer has confirmed orders for one million laptops (Nystedt & IDG News Service 2007). However, the laptops cost around 130$ and the 100$ price target has not yet been reached.

[459] Picture provided by OLCP (2007); licensed under the Creative Commons Attribution2.5 License.

[460] It is thought that in the United States of the 60s this point was between 20% and 30% (Pryor 1971: 413). However, for Birmingham in the United Kingdom Woods (1977: 488) found no particular tipping point.

(Gladwell 2000: 3-14; see also Levitt & Dubner 2005: 117-144).[461] Another example is the influence of individuals with high status (professional or managerial jobs such as managers or teachers) on school dropouts and teenage pregnancy (Crane 1991). School dropouts and teenage pregnancy but also other social problems often spread through peer influence and peer pressure (Crane 1991: 1227). If there are not enough individuals in the community able to handle the problems a chain reaction may start. In neighborhoods with a percentage of high status inhabitants between 5% and 40% there is little difference in school dropouts or teenage pregnancy. However, if the percentage falls below 5% the dropouts and teenage pregnancy increases dramatically and significantly (see Figure 72).[462] It therefore seems that the tipping point in which problems escalate is around 5% of individuals able to handle problems.

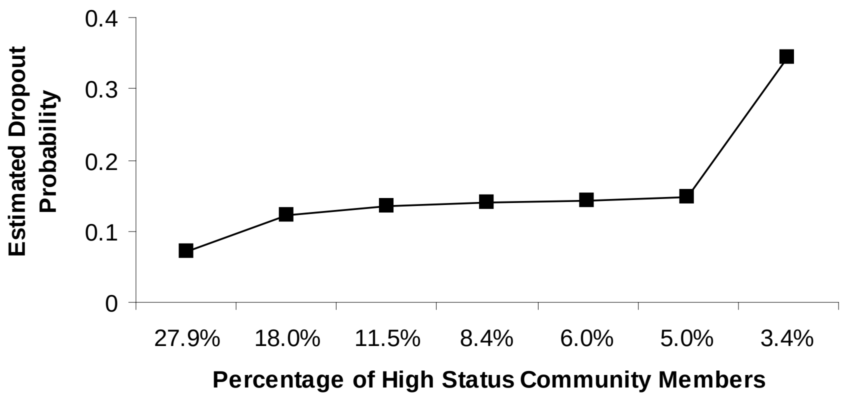


Figure 72: Estimated dropout rates among black males in large cities[463]

Yet another example of a tipping point is an epidemic of syphilis that attacked the city of Baltimore in the mid-nineties. Within a year the number of children born with the disease increased by 500% (Gladwell 2000: 15-19). One out of three explanations for this outbreak is that the city of Baltimore cut back the number of clinicians at the city's sexually transmitted disease clinic from seventeen to ten and the number of physicians from three to zero.[464] In consequence, the number of treated patients dropped from 36'000 to 21'000 a year. This decrease of treated patients changed the acute infection of the patients into a chronic disease. In consequence, the disease disrupted because the disease carriers' had three to four more time to pass on the infection. Thereby the disease got out of control. The cut back of only ten persons working in a specialized clinic seems to have heavily contributed to the disruption of the epidemic.

[461] Tipping points are somehow related to the broken window theory (Wilson & Kelling 1982). However, it does not seem to be clear whether this prominent theory of crime prevention has proved to be correct or not (e.g. see Harcourt & Ludwig 2006).

[462] The distribution for white and female individuals is very similar but less extreme.

[463] Crane (1991: 1256).

[464] All three explanations share the characteristic that the responsible reason itself was not a dramatic occurrence (Gladwell 2000: 17).

6.1.3 Threshold models

In threshold models individual behavior is Bernoulli respectively binary distributed (Wood & Doan 2003: 645-646). Individuals either act (1) or do not act (0) with no possible behavior between. A threshold model assumes that there is a change in the individual behavior after a certain threshold value has been crossed.[465] A threshold is defined as the minimal proportion of a population who must decide to act in a certain way before an individual does so as well (Yin 1998: 535). Each individual has a different threshold level. Some individuals act even if almost nobody does so as well while other individuals need almost an unanimous turnout to act.

The dynamics of collective action is contingent on the distribution of thresholds

In society there are different distributions of thresholds consisting of individual thresholds. According to the threshold model the dynamics of collective action depends critically on this distribution of threshold (Wood & Doan 2003: 641). The dynamics associated with threshold models are usually thought to be nonlinear. The nonlinear distribution of thresholds is responsible that small changes sometimes suddenly and dramatically change the preexisting condition (Wood & Doan 2003: 640). In most cases many individuals act only if a noticeable amount of individuals has already acted and there are therefore not many individuals with a low threshold. At the same time only few individuals need full participation to join a cause. In consequence, thresholds are often normally distributed.

If thresholds in a society are normally distributed the change from a tranquility equilibrium into revolution are especially dramatic (Yin 1998: 564).[466] In such a case at the beginning an individual joining the cause does not influence other individuals to participate because their threshold is not yet reached. However, at a certain point the participation of one additional individual leads to a participation turnout for which the threshold of many others is reached. In consequence, a cascade of participation starts. Societies with a bipolar distribution of thresholds are less stable but changes are less dramatic.

In societies in which thresholds are normally distributed a paradoxical situation may emerge in which almost everybody wishes to change the situation but nobody does want to be the first opposing voice and bear the related costs.[467] In such cases the costs for opposition are so high that individuals accept that their public positions do not match their private positions. However, remaining silent leads to tolerance costs. The tolerance costs, like the opposing costs, are for each person different. For a reticent individual tolerance costs may be lower than for an activist (Wood & Doan 2003: 642).

Granovetter's model

Granovetter (1978) studied a threshold model related to the propensity of individuals to join a riot. In his model there is a *heterogeneous* group of individuals with specific thresholds concerning the number of other's who must have chosen the alternative in question. In such a model every individual has the potential to push

[465] In toxicology threshold models posit that any dose above a certain level is dangerous and any does below is safe. Threshold models have long been used establishing acceptable exposures to non-carcinogens (Calabrese & Baldwin 2003: 246).

[466] Thereby the change is the more drastic the lower the variance.

[467] See in chapter 6.4.1 the story narrated by Vaclav Havel. It shows how such a paradoxical situation may last for a long time.

another individual over the threshold. This may induce a chain reaction going on until the next higher threshold of an individual is not reached.

The major implications of Granovetter's model are the following:

i. The number of individuals who choose an alternative increases until it stagnates at some point and it is therefore monotonically increasing.

ii. The few extreme individuals with very low thresholds are extraordinarily important because they moderate the behavior of individuals with an average threshold size. This is crucial because most individuals have an average threshold size.

iii. The actual outcome in a certain decision situation depends largely on the frequency distribution of thresholds.

There are several variants of Granovetter's theory (Chong 1991; DeNardo 1985; Kuran 1989, 1991). For Kuran (1991: 17-18) a regime is replaced if the opposition reaches a certain size. In his model the costs to join the protest movement decreases with its size. Extreme individuals start the movement and moderate individuals will follow. In the model of DeNardo (1985) the government has unlike in the theory of Granovetter the possibility to react to protest with reforms and in consequence reduce the size of the opponents. Chong (1991) not only discuss the interaction between protest movement and a regime but also includes a counter protest movement.

Granovetter's model discussed above is relevant to all situations in which individuals have to choose between two alternatives and the associated benefits depend on the number of other individuals choosing one or the other alternative (Lohmann 1994: 47). Therefore, it cannot only be applied to revolutions but also to other public goods. Other research based on threshold models include the persistence of India's caste system (Kuran 1995: 196-204), the decision to join the civil right movement (Chong 1991), and Arthur's (1989) description of lock-in to inferior technology. Another scholar who studied threshold models is Thomas Schelling (see also chapter 6.1.4). Schelling's (1978: 213-243) model is similar to the one of Granovetter. Some of his studies were related to the process of segregation of residents (Schelling 1971a; Schelling 1971b, 1973b). In his model the threshold for leaving a neighborhood is a function of how many individuals of one's own color have already left (see also chapter 6.1.2).

Who starts revolutions?

Marwell and Oliver (see Oliver, Marwell, & Teixeira 1985; Oliver & Marwell 1988, 2001) expanded Granovetter's theory. They argue that there are individuals who have a big interest in the outcome of collective action and therefore a strong incentive to contribute. These individuals may build a critical mass that is in turn the base for the contributions of other individuals. Heterogeneous groups enhance therefore the probability of collective action because the critical mass must be based on individuals with extremist tendencies (see also 6.3). Alexis de Tocqueville (1964: 139) observed that people with very low threshold show special characteristics; he pointed out that in revolutions madmen have played a considerable political role:

> *J'ai toujours pensé que dans les révolutions et surtout dans les révolutions démocratiques, les fous, non pas ceux auxquels on donne ce nom par métaphore, mais les véritables, ont joué un rôle politique très considérable.{I have always thought that in revolutions and particularly in democratic revolutions, madmen,*

not those to whom one gives the name metaphorically, but real ones, have played a very considerable political role.}

It can therefore be concluded that collective action is started by individuals with special characteristics and strong preferences.

Thresholds beyond human beings

Threshold models do not only apply to human beings but also to mass migrating animals.[468] With increasing density of animals in a group a rapid transition may occur from a disordered movement of individuals to a highly aligned collective motion (Buhl et al. 2003: 1402). Extremely interesting examples are locusts. The behavior and appearance of locusts may change due to density so dramatically that it was thought until 1921 that the locust *Schistocerca gregaria* in the ordered and unordered phase were two different species (FAO 2006).

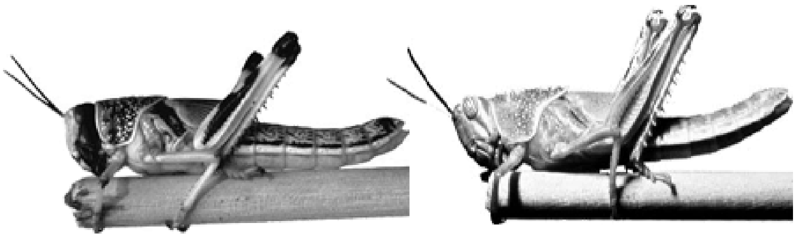


Figure 73: Schistocerca gregaria in the ordered (left) and unordered (right) phase[469]

Buhl et al. (2003: 1403) studied in labor experiments the behavior of this desert locust. If the density of locusts per m^2 is below 17.2 there is almost no coordination between the animals. In this solitarious phase the locust goes its separate way, is harmless, and does not threaten agricultural production. However, if the density increases over a threshold of 24.8 locusts per m^2 the animals start rapidly to align their directions of travel. In this phase the movement is coordinated but extremely unpredictable. The swarm changes rapidly and frequently its direction. If the density moves beyond 73.8 locusts per m^2 the swarm rests coordinated but its movement gets additionally predictable. The locusts are now in the actively aggregating, band-forming gregarious phase. One single swarm of this type may cover several hundred square miles with 40 to 60 millions locusts per square mile (FAO 2006). All swarms together spread over 20 percent of total land surface of the world and may destroy agricultural production in the affected countries completely.

6.1.4 Schelling on binary collective decision making

Thomas Schelling (1973a) introduced a convenient diagrammatic representation of binary collective decision making. The horizontal axis in Figure 74 represents the cooperation rate from 0% to 100% while the vertical axis represents the benefit for a certain action ranging from low to high. Contingent on this cooperation rate the BN line represents the payoff for a non-cooperator, the BC line represents the payoff for a cooperator, and

[468] For a discussion of the relationship of habitat loss and extinction threshold see Solé and Bascompte (2006: 171-213).

[469] Lindsey and Simmon (2002); Photograph courtesy Compton Tucker, NASA GSFC.

the AC line represents the average payoff for all involved individuals. The distance between the BN and BC line at a certain cooperation rate represents the costs (or benefit) of cooperation.

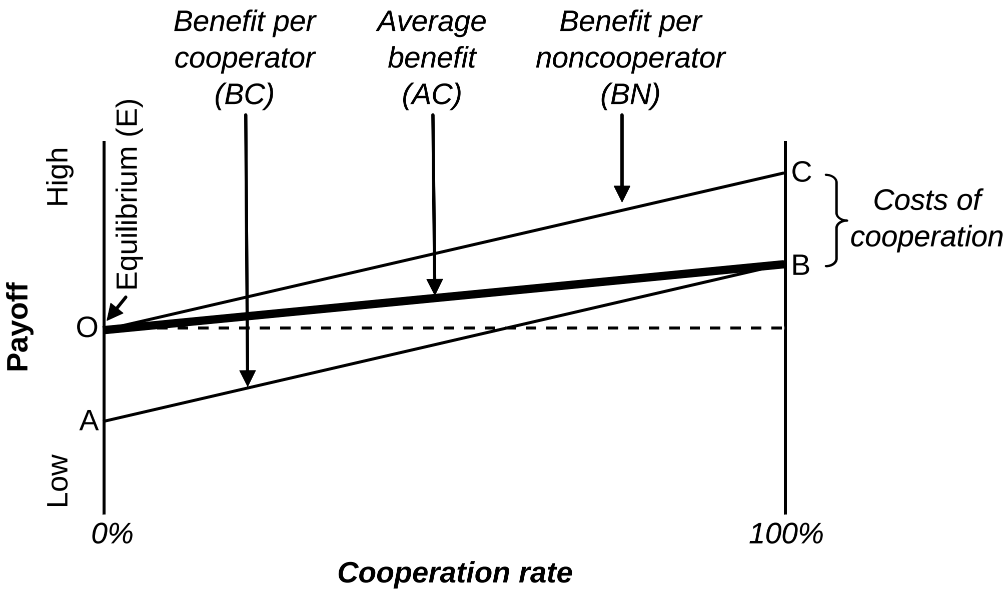


Figure 74: Many person prisoner's dilemmas[470]

The payoff for a non-cooperator depends on the cooperation rate. If nobody cooperates a non-cooperator receives in the example presented in Figure 74 the payoff zero. If everybody chooses to cooperate a non-cooperator *theoretically* receives the payoff C. Cooperators *theoretically* receive a payoff of A if nobody cooperates and the payoff B if every other individual cooperates as well. Payoffs A and C are only *theoretical* payoffs which are of course logically not possible.

In the simple case of the many person prisoner's dilemma the cost of cooperation (the distance between the two thin lines BC and BN) does not vary with the number of cooperators (see Figure 74). Non-cooperation is the dominant choice of every individual because the associated payoff is always higher than the one for cooperation. A collectively inefficient equilibrium (E) at point 0 is generated (Schelling 1978: 224-225). Unilateral cooperation (B) is more beneficial to everyone than unilateral non-cooperation (E).

That the cost of cooperation does not vary with the number of cooperators as assumed by the prisoner's dilemma is atypical (Elster 1989b: 128). The cost of cooperation may increase with an increasing cooperation rate (see Figure 75a). If people for example join a call-in campaign for a good cause the phone lines become congested and it needs more time to get through. Nevertheless, it is also possible that the costs of cooperation decreases with an increasing cooperation rate (see Figure 75b). If more people join a revolution the probability of being punished is getting lower.

[470] Adapted from Elster (1989a: 28) and Schelling (1978: 220).

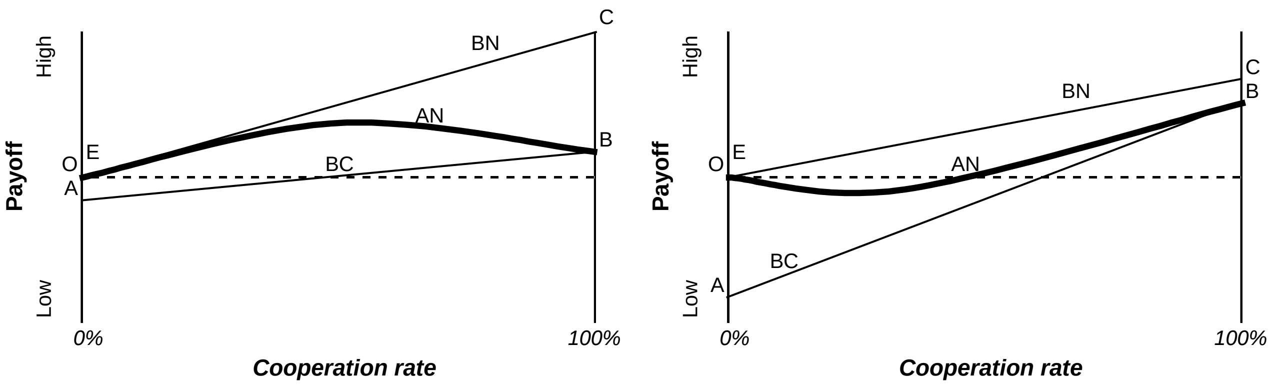


Figure 75ab: Increasing and decreasing cooperation costs[471]

It may also be that only for the first few cooperators non-cooperation is the dominant choice (Figure 76a). If the cooperation rate is increasing the payoff for cooperation may raise as well. In such situations there are two equilibriums. If the cooperation rate is lower than the one at the intersection of BC and BN nobody will cooperate while higher cooperation rates lead to universal cooperation. An example may be keeping a park clean. Everybody does profit if nobody is littering a park. The first individual who stops littering will have almost no effect on the amount of garbage in the park. This individual has to bear the (minuscule) costs of walking a few feet to the next trash can. Additionally, he or she has to bear the cost of being considered *a jerk green fundamentalist*. As we know now from the Asch conformity experiments (see Bond & Smith 1996) or Elisabeth Noelle-Neumann's (1974; Noelle-Neumann 1993) spiral of silence (see also chapter 6.2.2) most individuals do hardly oppose a clear majority even if there is no formal punishment for doing so. This may be explained by a preference to belong to a majority, by social disapproval, by social pressure, and the like. The costs of cooperation are then partly based on psychological costs. However, if more and more individuals stop littering the park the costs of cooperation may decrease. If many individuals refrain from littering social disapproval and social pressure against *jerks who litter* may emerge. After a certain turning point the costs related to such a social disapproval may be higher than walking to the trash can. The second equilibrium is then full cooperation.

In situations such as the one discussed above the question is how to get cooperation to the turning point respectively over the threshold from which cooperation is self-sustaining. One may provide selective incentives for the first cooperators or a Leviathan may make cooperation mandatory until the turning point.[472] An example is the hockey helmet story told by Schelling (1978: 213). The first player who wore a hockey helmet was considered being a wimp. Therefore, nobody wore hockey helmets voluntarily. At same time most players preferred to wear a helmet. In this situation the hockey league had to force the players to wear helmets.

[471] Adapted from Elster (1989a: 29) and Schelling (1978: 220).

[472] Certainly, a practical and in many cases unresolved problem is to know the exact turning point.

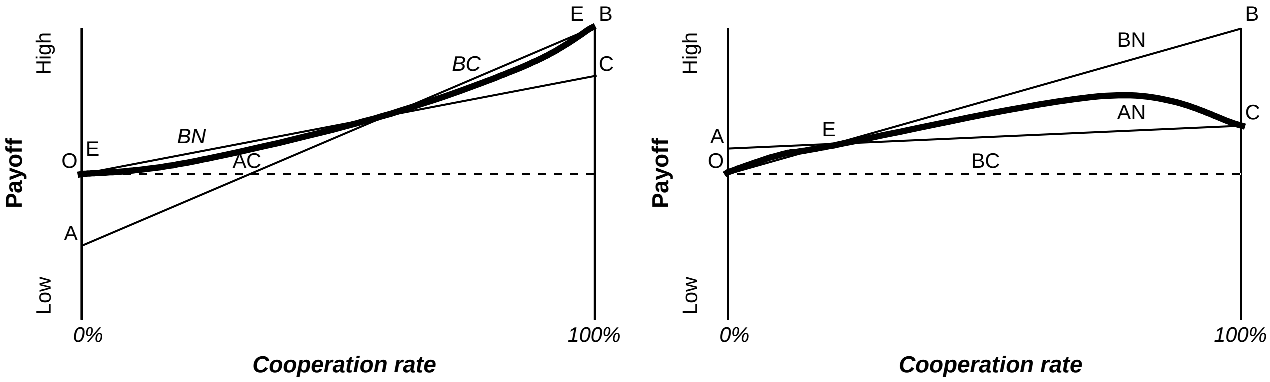


Figure 76ab: Equilibriums in collective actions[473]

It is not imperative that the first cooperative move is more costly (see Figure 74, Figure 75, Figure 76a, and Figure 77b) than a non-cooperative move. It may be that the first individual initiating cooperation gets credits for it (payoff A, see Figure 76b). This reputation benefit may fade out very quickly and the costs of cooperation become higher than the associated benefits. In such a case there is only one equilibrium (E) at the intersection of BN and BC.

In the examples above the payoff for a cooperative move (BC) was always increasing with ongoing cooperation. This is not necessarily the case (see Figure 77a). It may be that the first cooperators actually profit from their decision but that this profit gradually degrades until cooperation starts being costly. It is also not imperative that the costs of cooperation change monotonically (see Figure 77b). In the example presented in see Figure 77b the cooperation costs decrease until the first intersection of BC and BN from which on cooperation gets beneficial and is the dominant choice. With the cooperation going on the costs of non-cooperation decrease and after the second BC/BN intersection it is more beneficial not to cooperate. From this point on it is more costly to cooperate than to non-cooperate. In the related example there are two equilibriums (E) at point 0 and at the second BC/BN intersection. Either nobody will cooperate or if the start-up problem is solved (about one third must cooperate) a chain reaction starts until about two thirds of the individuals cooperate.

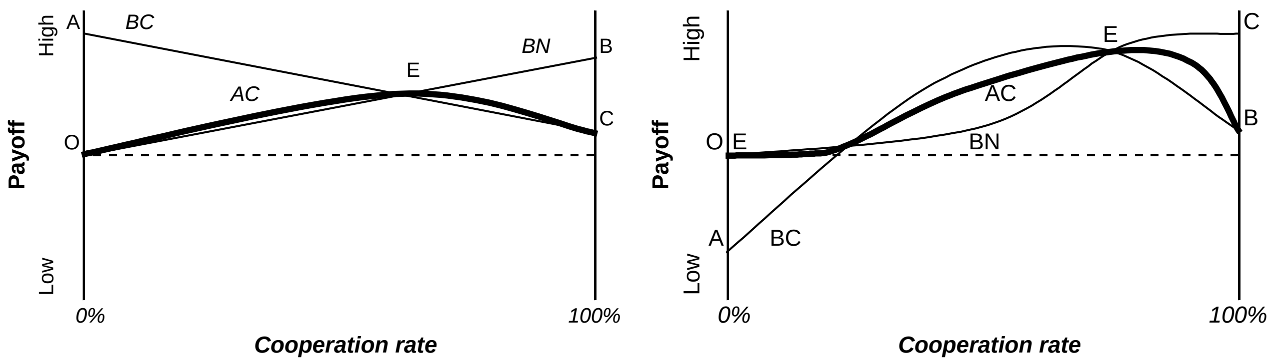


Figure 77ab: Changing cooperation costs/benefits[474]

[473] Adapted from Schelling (1978: 228-229).
[474] Adapted from Schelling (1978: 229 and 240).

The reason for discussing the dynamics of collective action is above all to show that cooperation costs may change contingent on the cooperation rate. It is argued that the many person prisoner's dilemma with constant cooperation costs is a special case. Contingent on the cooperation rate the costs of cooperation may decrease. In such cases, it may be beneficial for an individual to cooperate as well after a certain cooperation rate. Being the very first revolutionist and being therefore the first cooperator may be very costly. However, being the very last apparatchik and being the very last non-cooperator may be very costly as well. Schelling's diagrammatic expression of collective action shows also that it may be important to get over a tipping point.

6.1.5 Increasing return technologies

The characteristics of increasing return technologies (Arthur 1989: 116) introduced in chapter 2.9.3 are very similar to critical mass, tipping points, and threshold models. This can be pointed out with a simple example presented by Arthur (1989: 119): Individuals can choose either technology A or B (see Figure 78). The first agent with no predecessor chooses the more favorable technology A for which the payoff is 10. This is more than twice the payoff for technology B. By choosing A the first agent enhances the returns for the second agent for adopting A. After 20 agents have chosen A the related payoff for doing so is three times higher choosing technology B. Because of this increasing difference in payoffs it gets more and more costly to choose technology B. The market is locked in. However, this is in the long run not efficient. Technology B is after 30 adoptions more efficient. For the first agents adopting a technology choosing B is collectively but not individually rational. With the time the conflict between collective and individual rationality dissolves. Therefore, the collective action problem gradually gets smaller and finally comes to nothing. In this situation the crucial question is how to solve the start up problem and to get over the tipping point of these 30 adoptions to foster the superior technology B. It needs a critical mass of individuals using technology B before its superiority will lead the further advancement automatically.

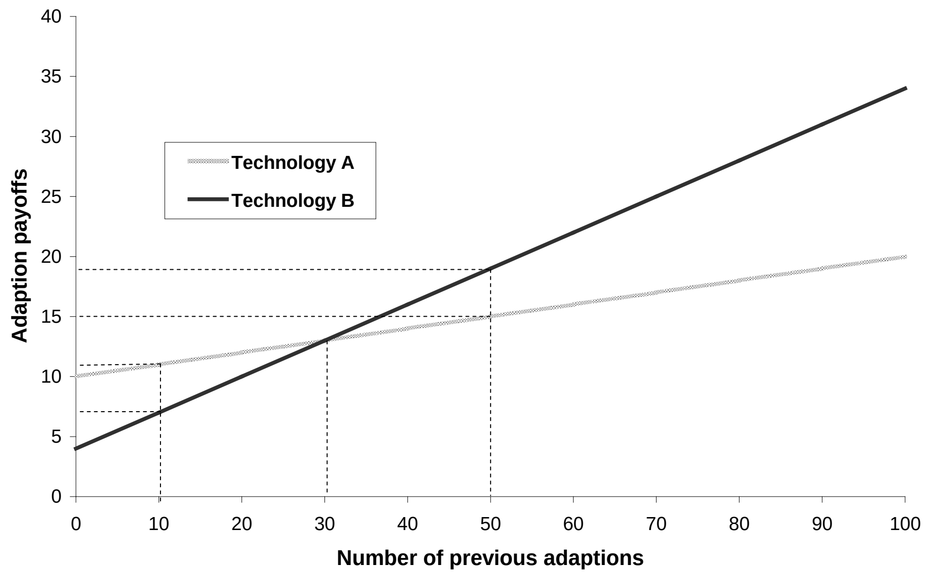


Figure 78: Adaption payoffs for different technologies[475]

[475] Arthur (1989: 119).

6.2 Incremental decision making and herding effects

In the following, the power of incremental decision making is highlighted. *Firstly*, it is exemplarily shown how incremental decision making may lead to a degeneration of fortune. *Secondly*, it is exemplified that fortune may also increase by trading incrementally. Afterwards herding behavior and bandwagon effects will be discussed.

This subchapter intends to highlight the basic idea that some occurrences can only be explained by incremental steps or incremental decision making. This is presented with descriptive stories and descriptive concepts.

6.2.1 Hans im Glück forwards and backwards: The power of deciding incrementally

Trading goods incrementally can degenerate the fortune as it is reported in the fairy tail "Hans im Glück" (Hans in Luck) of the Brothers Grimm (1812).[476] After having served seven years Hans decided to leave his master and go back home. Because he served faithfully and honestly his master gave Hans a piece of gold as big as his head. On the way home he first meet a horseman with which he traded the piece of gold for the horse. Afterwards he traded the horse for a cow, the cow for a pig, the pig for a goose, and finally the goose for grindstones. Carrying the heavy stones made him thirsty and he refreshed himself at a well in a field. To do so he laid the grindstone by his side on the edge of the well. Accidentally, he pushed against the stones, and both of them fell into the water and were lost.

What makes the fairy tale Hans in Luck especially interesting for this dissertation is that during his way home he *incrementally* lost more and more of what he got for his seven years of work. He probably would not have traded the piece of gold directly for the grindstones. However, trading a horse for a cow may be reasonable as one can produce milk, butter, and cheese every day that may be more valuable than riding a horse. Similarly, trading a goose for grindstones is not necessarily a bad deal. That Hans traded very badly is therefore not apparent if only the single trades are examined but very clear if the first and the last good are compared. The possibility of making incremental decisions may therefore change decision making tremendously (see also sequential decision making in chapter 5.4.2.1.1).

However, acting incrementally does not necessarily lead to degeneration of fortune. It also may be that by incremental decision making fortune can be increased. An impressive example of Hans in Luck backwards is the story of Kyle Donald who incrementally traded within a year one red paperclip for a two-story house (BBC News 2006b). He traded first the paperclip for a pen shaped like a fish. With the time going on he could trade for a handmade doorknob, for a camping stove, for a 1,000-watt generator, for a beer keg with neon sign, for a snowmobile, for a trip to Yakh, for a large van, for a recording contract, for a one-year rent-free in a bungalow

[476] The statement that the value degenerates only refers to the material value. Hans seems to be very happy after having lost everything: He says finally: "So glücklich wie ich gibt es keinen Menschen unter der Sonne" (There is no man under the sun as fortunate as I). This seems to indicate a happy end (Uther 1990). However, there is disagreement about whether Hans is lucky or not (Sahr 2002: 92-110). While some argue he is not Ludwig Marcuse (1972: 42-49) prominently argues that Hans is the first philosopher of happiness and that he is indeed lucky. Some such as Janosch have put the fairy tale to the extreme and continued the story after he lost his grindstones ("ein Bein verloren, eine Schlampe geheiratet, aber dennoch kreuzfidel" {lost a leg, married a bitch, but still happy as a king}) which makes it hard to belief that he is still lucky (cited in Sahr 2002: 100). After all, the question whether Hans is lucky depends on the definition of happiness. The ambiguity about happiness seems to be one of the factors for the tremendous success of this fairy tail.

in downtown Phoenix, for an afternoon with Alice Cooper, for a snow globe[477], and for a role in a Hollywood movie. Finally, Donald traded the movie role for the house. Apparently, nobody would have traded a house for the paperclip directly. Donald could only manage to get the house by incremental steps. Again, this impressively shows the influence of incremental decision making.

6.2.2 Herding matters

Herding behavior describes how individuals of a group act together without having a planned direction. It is based on the behavior of animals in herds, flocks, or schools. However, herding behavior is also observed for human beings. It occurs for example frequently during the evacuation of a crowd in room with two exits (Xiaoshan 2006: 38). Even if both exists are functionally similar it is often observed that one exit gets highly congested while the other is not fully utilized. The primary parameter for herding behavior is perceived uncertainty (Xiaoshan 2006: 39). If individuals have *insufficient* information and do not know what to do, they tend to follow the crowd.

Herding behavior respectively doing what everyone else does even if *some* private information suggest doing otherwise (Banerjee 1992: 798) is also evident in financial markets. In his famous beauty competition analogy Keynes (1936: 156) states that in stock markets individuals do not care what they think is the fundamental value of a share but rather what they think everyone else think about the value of the share. Scharfstein and Stein (1990: 465) argue that under certain circumstances "managers simply mimic the investment decisions of other managers, ignoring substantive private information".

Following the crowd may be quite rational. Security expert Schneier (2004: 119) argues that security in cryptography „comes from following the crowd".[478] A cryptographic algorithm is only secure if it is challenged again and again. This is only done for widely used algorithms.

The dynamics of collective action exhibit herding effects (Lohmann 2000: 656). Herding behavior occurs in collective action if people's decisions are interdependent over time (Lohmann 2000: 657). Such interdependencies rely on payoff externalities in which each individual's benefits and costs of an alternative depends on the number of other people who do so concurrently or did so cumulatively over the past (Becker 1991; Granovetter 1978; Kuran 1989, 1991; Marwell & Oliver 1993; Schelling 1971a; Schelling 1973a, 1973b, 1978; Young 1996). Lohmann (2000: 658) states:

> *The aggregate number of people adopting the behavior in each period or cumulatively over time then follows from individual incentives to adopt the behavior, which in turn are determined by the aggregate number who adopt the behavior, and so on.*

Herding behavior influences public opinion very strongly. Elisabeth Noelle-Neumann (1974; 1993) coined the term of the spiral of silence.[479] Basically, it states that an individual's willingness to express his opinion about

---

[477] Interestingly, the snow globe seemed not to have much value. It was merchandise from the band KISS that could change light with variable speed and looked somewhat cheap. However, a snow globe collector was eager to trade for it.

[478] Schneier (2004: 119) states that „a homegrown algorithm can't possibly by subjected to the hundreds of thousands of hours of cryptanalysis that DES and RSA have seen. A company, or even an industry association, can't begin to mobilize the resources that have been brought to bear against the Kerberos authentication protocol or IPSec. ... By following the crowd, you can leverage the cryptanalytic expertise of the worldwide community, not just a few weeks of some unnoteworthy analyst's time.„

a *value-laden issue* is a function of how he perceives public opinion (Scheufele & Moy 2000: 3) that is heavily influenced by mass media (Scheufele & Moy 2000: 11). Individuals fear of being isolated (Scheufele & Moy 2000: 9). This spiral of silence is a dynamic phenomenon (Noelle-Neumann 1974: 44):

> *… the tendency of the one to speak up and the other to be silent starts off a spiraling process which increasingly establishes one opinion as the prevailing one.*

6.2.3 Bandwagon effects

In a known descriptive classification Leibenstein (1950: 188) distinguished different types of consumer demand for goods and services. While Leibenstein did not discuss the determinants under which one type of demand prevail over the other he showed how different types of demand might shape events. He basically distinguished between functional and nonfunctional demand. Nonfunctional demand was subclassified further in three different subcategories (1950: 188):

A. Functional demand

B. Nonfunctional demand

    1. Demand based on external effects on utility[480/481]

        (a) Bandwagon effect

        (b) Snob effect

        (c) Veblen effect

    2. Speculative demand

    3. Irrational demand

Functional demand refers to the demand for a commodity that is due to the qualities inherent in the commodity itself. For nonfunctional demand a portion of the demand is due to factors outside these qualities (Leibenstein 1950: 189). The most important type of nonfunctional demand is due to external effects on utility. In such cases the utility of a good or service is enhanced or decreased because other individuals purchase or consume the same good or service as well. This contradicts standard theory on consumer demand in applied research which assumes that individual preferences and demand functions are independent (Chao & Schor 1998: 108-109; Leibenstein 1950: 184; Morgenstern 1948: 175). Demand is primarily understood as a

---

[479] She researched the willingness of individuals to discuss topics such as abortion law, capital punishment, or unmarried couples living together (Noelle-Neumann 1974: 46).

[480] Logically, a counter-Veblen effect is missing (see also Lea, Tarpy, & Webley 1987: 204-205).

[481] This family of anomalies differs from Giffen goods which also show an interaction effect between price and demand. People buy more of a Giffen good if a price rises. However, Giffen goods are the result of income and substitution effects. The classical example of Marshall (1910: 132) is inferior staple food like bread whose demand is driven by poverty. If the price of cheap staple food rises poor people can no longer afford superior food and most consume more of the staple food. The preconditions for a Giffen good are that i) the good must be inferior, ii) there is no close substitute good, and iii) the good must encompass a substantial percentage of the buyer's income. Samuelson (1967: 423) states that in the Irish famine of 1845-1849 potato was a Giffen good. Conversely, many authors question the validity of the potato example (Dwyer & Lindsay 1984; Rosen 1999). Jensen and Miller (2002) argue that in the north of China noodles and in the south of China rice are Giffen goods. Some authors suggests that Giffen goods are empirically irrelevant (Marshall 1910: 132-133; Stigler 1947: 156, 1987: 23), difficult to observe (Nachbar 1998), or pure fiction (Rosen 1999: 313).

relation between individuals and goods (see also Bergstrom 1970: 383). In consequence, collective demand is generally understood as a mere summation of individual demand schedules (Morgenstern 1948: 175). Status consumption and other actions motivated by an interpersonal comparison are ruled out. However, Chao and Schor (1998: 128) found strong support for the existence of status motives in the purchase of women's cosmetics. They conclude that without invoking social factors it is difficult to explain why consumers pay $181 an ounce for a lipstick that is intrinsically not different from one for $18 an ounce. The collective demand curve of "snobs" is most likely not additive (Leibenstein 1950: 183). Leibenstein differentiated demand based on external effects on utility, as described above, into bandwagon, snob, and Veblen effects. In the following bandwagon effects will be discussed in length because it is thought to be relevant for the advancement of this dissertation. For the sake of completeness Snob and Veblen effects will be briefly discussed.

Veblen effects

For a Veblen good demand increases as a direct function oft its price (Leibenstein 1950: 189). Such goods are related to Veblen's (1975: 68-101) concept of conspicuous consumption that is used for displaying income or wealth and in turn may serve as a mean of attaining or maintaining social status. Status is a positional good that is in fixed supply and any conspicuous consumption produces negative externalities. Individuals become subject to invidious distinctions that form the conventional basis of esteem and self-respect (Veblen 1975: 28-29). Consumption for the purpose of gaining status must hold *firstly* the precondition that individuals must share some degree of commonality in their ranking of the relative desirability of products or services and *secondly* the consumption must be socially visible and therefore externally verifiable (Chao & Schor 1998: 111). Therefore not astonishingly Chao and Schor (1998: 121) found systematic differences across products by the level of social visibility. High status goods like wines or perfumes or the aforementioned lipsticks may belong to this category of Veblen goods. Another example of a Veblen good may be the "I am Rich" application for the Apple iPhone. The only function of the application is to display a glowing red jewel on the iPhone screen and indicating wealth because it costs $999.99.[482] The official description of this application was (Wagenseil 2008):

> *The red icon on your iPhone or iPod Touch always reminds you (and others when you show it to them) that you were able to afford this. It's a work of art with no hidden function at all.*

Snob effects

Snob goods are based on the desire for exclusive or unique goods (Leibenstein 1950: 189). Therefore, a snob good is in contrast to a Veblen good not a function of price but of the number of consumers.[483] Examples are rare works of art or designer clothing. Collectors stamps or coins that are the rarest are the most valued. The 1-cent postage stamp "Benjamin Franklin Z-Grill" issued by the United States Postal Service in 1868 in not only the rarest of all United States stamps but also the most expensive (Schmid 2006).

---

[482] Actually, if one touches an information button the following message appears additionally (Anonymous 2008, sic!): „I am rich I deserv it I am good, healthy & successful".

[483] However, Veblen and snob effects are to some degree related to each other.

Speculative and irrational demand

Speculative demand refers to the observation that individuals buy commodities because they expect the price to rise and earn money by selling these commodities (Leibenstein 1950: 189). Irrational demand refers to purchases that are neither planned nor calculated but are due to sudden urges and whims.

Bandwagon effects

For a bandwagon good the demand increases if others consume the same good as well. Such goods refer to the desire to get into “the swim of things” or to be “one of the boys” to conform to other individuals (Leibenstein 1950: 189).[484] Such bandwagon effects do prominently exist for culture and arts. The admiration of certain pieces of art and culture is to some extent self-reinforcing (Rohner, Winestein, & Frey 2006). If a critical mass of individuals has decided to consume a piece of art others do as well. It is claimed here that many people want to see the painting Mona Lisa in the Musée du Louvre because others have seen it too. The Musée du Louvre houses 34’999 other works (Musée du Louvre 2007) that are probably similar or even more interesting.

The literal meaning of a bandwagon is a decorated wagon that carries musicians in a parade (Rohlfs 2001: 1). Bandwagons were used in elections in the United States to mobilize voters. Jumping on the bandwagon and take a ride was so popular because one could enjoy listening to the music without having to walk. By and by people followed the bandwagon because others did as well. In politics bandwagons were first used by the famous clown Dan Rice (Carlyon 2001: 78). He campaigned for Zachary Taylor as President of the United States and invited him to ride on his circus bandwagon. Today jumping on the bandwagon is a phrase for joining an increasingly popular trend.

Concerning bandwagon effects a huge amount of literature discussed the question whether pre-election and exit polls influence the result of a ballot (Bartels 1987; Ceci & Kain 1982; Fleitas 1971; Gallup & Rae 1940; Hardmeier & Roth 2003; Marsh 1985; Mcallister & Studlar 1991; Milavsky 1985; Nadeau, Cloutier, & Guay 1993; Navazio 1977; Simon 1954; Skalaban 1988; Sudman 1986; Vowles 2002). According to Herbert Simon (1954: 246) such a bandwagon effect occurs “if persons are more likely to vote for a candidate when they expect him to win than when they expect him to lose”.[485] The influence of polls on the result is the reason why they are sometimes forbidden if they are too close to the polling day or polls are still open (Milavsky 1985; Vowles 2002).

An example of a bandwagon effect is the invasion of pogs in French primary schools in the 90s (Corneo & Jeanne 1997: 335-336).[486] Pogs are small plastic discs with little intrinsic value. They can be used for a very simple game in idle times. French children developed a strong taste for collecting these pogs. Original pogs had a much higher value than the similar products from other producers. The status of a child among his peers was very much influenced by the number of *true* pogs. Everybody familiar with the evolution of such schoolyard games may probably guess how the pog bandwagon may have developed. First, a pupil with high status and connections to pupils from a bigger town appears with the game. Everybody thinks that the game is

[484] The actual effect on demand rests on the range of other goods, their prices, and on the degree of substitutability of the good in question.

[485] Bandwagon effects based on pre-election polls are usually contrasted to underdog effects. Underdog effects occur if voters support a candidate because he seems to loose. While such an effect seems to be quite important in the political sphere it does not seem to play a significant role for technology and FOSS development.

[486] It is not hard to find similar examples such as the Panini stickers sold during soccer World Cup.

somewhat strange but curiosity drives friends of the pupil to buy the game too. With the time going the favorite activity in idle times is playing this game. Pupils with no game cards, disks, or whatever are mavericks and therefore struggle to get the game too. Finally, everybody (at least of the same gender) has the game. Afterwards the collecting behavior collapses very fast and almost immediately.

The largest and most successful bandwagon apart from the telephone is probably the Internet (Rohlfs 2001: 5). It gained a tremendous momentum although it faced severe problem at its startup. It was expensive and one could only communicate with a very limited number of users. At some point the internet bandwagon took speed on and its interlinking accelerated this process (Rohlfs 2001: 191-192). The example of the Internet shows also that the direction of the bandwagon is sometimes coincidental. If the University of Minnesota had not announced to charge license fees for the use of gopher servers the Internet would nowadays perhaps not be dominated by the World Wide Web but by the gopher space.

Solving the start-up problem and reaching a critical mass is essential for success in bandwagon markets (Rohlfs 2001: 196). After this critical mass is reached the demand for a product or service is subject to positive feedback respectively to herding mechanisms and may grow extremely rapidly. A possibility to solve the start-up problem for an electric appliance is to have a stand-alone solution that is even valuable without network effects (Rohlfs 2001: 197).[487] An example is the videocassette recorder. While there are clear network effects if cassettes can be exchanged or rented the recorder is also valuable as a stand-alone product. Using video recorders for watching television time-shifted does not depend on network effects.

Bandwagon effects are related to the rhetoric principle “argumentum ad populum” (appeal to the people) which states that something is true because many others believe it to be true (Walton 1989: 84). In advertisement it is sometimes shown that everybody has product X. This indicates that one is a maverick if one does not have it. It also implies that so many individuals can hardly be wrong and therefore the product must be superior. Inferential statistics follows actually a similar line of reasoning. If it is observed that smokers get more often lung cancer one infers that smoking causes cancer. In politics this line of reasoning is also often found. Candidates who run for office declare often to be part of whatever the majority is or does. In the announcement of Hillary Clinton to form a Presidential exploratory committee to run for Presidency in the 2008 election she did not forget to mention that she “grew up in a middle-class family in the middle of America” (Clinton 2007). Similarly, during an Ohio senatorial race an opponent attacked Senator Robert A. Taft by saying that he is in contrast to him “from the common man” (Aldisert 1997: 185).[488]

A similar mechanism is expressed in the Chinese proverb "Three men make a tiger" (Loo & Yung 1997: 69; Wikipedia 2006e). In this story a Chinese advisor asked his king whether he would believe a person reporting that a tiger rambles in the town market what the king denied. The advisor asked what he would think if two men were to report it. The king stated that he would start wondering whether it was true. If three persons were to report it the king stated that he would believe it. The advisor told the king that the truth of an incredible story like a tiger in the market does not change because more people repeat it. He asked the king

[487] Bandwagon effects are in fact „one of the earliest and most original treatment of network effects (although that term was not yet in use)“ (Liebowitz & Margolis 2005: 449). While bandwagon effects are positive network effects snob and Veblen effects are negative network effects. For a discussion of network effects see chapter 2.9.3.

[488] Martha Taft, the senator’s wife, responded “My husband is being accused of not being a common man. That’s true. He was first in his class in college, first in his class in law school; and first in the U.S. Senate. You don’t want a senator who is common. You want someone who is outstanding. And that’s my husband!” (Aldisert 1997: 185).

therefore not to listening to his enemies while he is away what the king thought to be reasonable. In fact, when the advisor returned the king did not trust him anymore.

Informational cascades

To assume that reports about a tiger in the market are correct if only enough individuals confirm it respectively following the crowd may be quite rational (see also Young 1996). To vote for a second-preferred candidate only because everybody else tells to do so may be rational. This candidate has a real chance for wining the election. This motivation not too waste the own vote is a very strong drive that only few candidates run for office or only few candidates get votes above a minimum.[489]

That following a popular trend is sometimes rational can be explained by informational cascades (see Anderson & Holt 1997; Bikhchandani, Hirshleifer, & Welch 1992). If decision making is dominated by informational cascades every subsequent individual makes the same choice as individuals before independent of the own private signal. This can easily be captured by the following example (adapted from Wikipedia 2006b): Let us assume that there is a crossroad where a group of hikers must pass. The hikers have to choose whether to walk left or right. If a hiker chooses the wrong way he knows that he will be eaten by a grizzly bear. If he takes the proper way he will be save. The hikers have some information where the grizzly is but the information is imperfect. The hikers know that their best guess will be only right in two-thirds of the time. Alice who is the first to make a decision thinks that going left is the proper way. Because she has only her private information she will go left. Let us assume that Bob, the second hiker, will also think that the left way is correct. His private information as well as the information from Alice' decision will let him urge to go left. Now, even if hiker Carol thinks that right is correct she should go left anyway. The information that Alice and Bob took the left way is more convincing than her private information. In fact, the likelihood that both Alice and Bob are wrong is smaller than that her own information is wrong. All the hikers at the crossroad will from now on go left based on the decisions of Alice and Bob. This example shows that to take the other's decision into account is rational. However, if the game is played onward from time to time both Alice and Bob make a mistake and all hikers will be eaten by the grizzly.

The grizzly example above (as well as most informational cascade examples) is, of course, somewhat flawed. *Firstly*, in the example above the hikers follow the crowd because this reduces the likelihood that they make a wrong decision. However, the decisions of the previous hikers do not only change the probabilities but likely also the situation itself: Ivan, the ninth hiker in succession, can assume that even if going left is wrong and the grizzly waits at the left side there will be no grave consequences. After having eaten eight hikers the grizzly has had most probably enough. It is therefore very dangerous to not follow the crowd and go right even if the individual had almost perfect information that the grizzly waits on the left side. Going right may have the consequence that Ivan faces a very hungry grizzly. Based on the private information going left increases the

[489] The drive towards two opponents in plurality voting systems is also related to the median voter theorem. Mueller (1989: 182) writes that "… neither indifference {if the parties are too close together} nor alienation {if a party is too far away to make voting attractive}, nor the two combined will affect the tendency of two candidates to converge on the position most favored by the median voter when the frequency distribution of voter preferences is symmetric and unimodal" (see also Downs 1957: 115-141; Hotelling 1929). However, it is questionable if these theoretical considerations hold true in reality. As Hirschman (1970) argues not only exit but also voice may be influential. It may be that voters of the far left or the far right more likely use vote and not exit as a mean of influence (Kuster 1999). The assumption of symmetric and unimodal distribution of preferences may be too simplistic.

probability for Ivan meeting the grizzly but the consequences of meeting the beast are tremendously less severe.[490] The same mechanism has already been discussed in chapter 2.9.3. Following the crowd may be wise because there are network effects. If all others choose an inferior communication technology it is wise to do so as well even one has perfect information or knowledge that the technology is inferior. Therefore, the action of individuals does not only provide *information* but does also *change the situation* itself.[491]

Secondly, in the example above the individuals are considered to be homogenous as it is mostly assumed in literature on informational cascades (see however Smith & Sorensen 2000). However, even monozygotic twins have no similar personal traits (Jang et al. 1998: 1559). It can therefore be assumed that depending on their personality some hikers prefer either to follow always their private information or to follow always the crowd. It may be that some hikers have a very strong preference following respectively not following the crowd (see chapter 6.3). If there are two subsequent hikers who have a very strong preference to do *otherwise* than the majority the following hikers may be fooled if only the behavior of the nearest three hikers can be observed. The position of hikers within a group is also likely not equally distributed concerning physical characteristics. As probably many hikers have experienced the individuals in front of a hiking group are different from the ones at the end. The ones in front are physically the most fittest. They have thereby the highest chance to run away if they face the bear.[492] Because they know that their curiosity will probably drive them to face the grizzly on purpose. The later hikers not able to run away but following the crowd will pay for this decision. However, the later hiker could anticipate this and so on. This makes clear that the structure of informational cascades in real-life is not easy to grasp.

Bandwagon effects and the like in FOSS development

In the following, it is shortly examined whether the phenomenon discussed above is also, on a descriptive level, observable in FOSS development.

From a series of interviews with hackers of the Linux and Apache community Egyedi and Wendel de Joode (2003: 91) observe that something similar to a bandwagon effect also exist in FOSS development. Popular FOSS projects tend to become even more popular. A subject in the interviews stated (Egyedi & Wendel de Joode 2003: 91):

---

[490] A corresponding real story is told by Chamley (2004: 1). Penguins have to go fishing in the water. However, orcas might wait in the water to hunt for penguins. In consequence, no penguin wants to be the first jumping into the water. The probability for the first penguin being eaten is considerably higher than for the late followers because the orcas have at some point eaten enough penguins. In consequence, every penguin waits until another solves the start-up problem. At some point, a penguin is so hungry that he bears the cost of being the first. The myth around the building of the "Devil's bridge" in the Schöllenenschlucht in Switzerland which is en route to the St. Gotthard alpine pass is related to the example of Chamley. Because it was too difficult to build the bridge the devil was asked for help. The devil demanded in exchange for the help the soul of the first to pass the bridge. However, after the bridge was finished nobody wanted to cross the bridge that made it of course useless. Finally, a flaw in the pact with the devil was found and a goat was sent over the bridge. Thereby the start-up problem was solved. Being betrayed the devil unsuccessfully attempted to destroy the bridge with a gigantic stone that can be still visited.

[491] See for example the high definition optical disc format war between the Blu-ray Disc and HD DVD optical disc from 2005 to 2008.

[492] The example is certainly flawed. Even the fastest man on earth does not manage to outrun a grizzly. The grizzlies at the Yellowstone National Park are thought to have a maximum speed of at least 30 miles per hour and at least 25 to 28 miles per hour on average over two miles (Kearns 1937). This clearly outperforms the average speed of Asafa Powell and Justin Gatlin during their world record sprints for 100 meters (IAAF 2007). Therefore, the grizzly in the example must be known to be very old.

It is like a chain reaction, popularity will lead to more users, more users will lead to more developers, to more applications and thus to more users, etc.

Based on their qualitative approach Egyedi and Wendel de Joode (2003: 91) found three main causes for what they call the "bandwagon mechanism" in FOSS development:[493]

i. ***Getting reputation to produce a signal for potential employers:*** The desire to enhance the reputation is an important motivation to contribute to FOSS projects. To gain as much reputation as possible reputation investors choose projects with a large audience. They jump only on bandwagons that are already in motion.

ii. ***Reputation of project member is a signal for excellence:*** A former Debian project leader states that "when you see a good name, you will take a look". By doing so developers may stay within the project and start contributing. FOSS projects with highly known members will therefore attract additional developers. Reputation does then not serve as a signal for potential employers but for excellence. Most probably investors in human capital or fun seeker are attracted by beautiful source code written by somebody with good taste (see footnote 230).

iii. ***Download statistics as a signal for high quality software:*** In many areas there are countless FOSS applications targeting the same kind of user. For a regular user it is difficult and costly to evaluate which is the best solution. Download statistics is a signal indicating the quality of software. It could be assumed that software has been heavily downloaded because of its superiority. It can also be assumed that a project with many users will detect bugs sooner than projects with few users.

Information Week headlines such as "Sun jumps on open-source database bandwagon to boost Solaris" (Babcock 2005)[494] or Business Week statements such as "large companies are jumping on the Linux bandwagon for servers" (Ante et al. 2003) are further hints that bandwagon effects are important for FOSS. However, there are also negative bandwagons in FOSS development. Perens (1999b: 187) writes:

Caldera has had trouble keeping sufficient resources allocated to COAS to finish its development, and other participants have dropped off the bandwagon due to the lack of progress.

As we now know the Caldera (now SCO Group) bandwagon has completely stopped and the corresponding Linux distribution Caldera OpenLinux is defunct.[495]

Probably one of the most interesting occurrence related to commercial firms jumping on the FOSS bandwagon has been Sun's decisions to put Java technology under the GNU GPL (Sun 2007).[496]

---

[493] Actually, Egyedi and Wendel de Joode (2003: 91) state that the "bandwagon mechanism" can be explained by two different causes. However, they mix cause i) and ii) which does not seem to be appropriate because they are based on different motivational and theoretical assumptions. The former argumentation is based on reputation investors while the later is based on investors in human capital or fun seekers.

[494] Sun seems to have a penguin problem as Business Week called it (Kerstetter 2003). The success of Linux appears to be at the expense of other Unix operating systems such as Sun's Solaris operating system. Therefore, Sun's jump on the FOSS bandwagon and its decision to put Solaris and Java (partially) under FOSS licenses seems to be more a desperate move than a belief in FOSS principles.

[495] The success of the lawsuits of SCO against IBM and Novell is evaluated differently. Groklaw (2006b) claims that "SCO is toast". SCO itself state in a letter of the President and CEO to customers and partners that they "have had a few setbacks in the court proceedings" but "important and significant claims remain in the case" (McBride 2007). The letter paraphrases a line from Twain that "the rumors of our death have been greatly exaggerated".

6.3 Mixed motivations, group heterogeneity, and the dynamics of collective action

That the composition of individuals with a certain motivation or background is subject to timing effects can be recognized by the migration patterns within urban areas. Students and artists move into an abandoned area because rental fees are low and informal rules are tolerant. After some years the town districts gets more and more lively and attracts yuppies. Grocery shops are replaced by bars or cloth shops, prices rise, and rules get fixed. That is why the original residents move out or have to move out and look for another abandoned area. Thereby the area gets boring, yuppies move out as well and the story starts again. In the following, a similar principle will be discussed in length.

Elster's chain of collective action

According to Elster (1989a: 187) successful collective action is based on a mix of different motivations. He argues that taken separately the motivations would lead not to successful collective action (Elster 1989b: 131). The different motivations interact and build upon each other so that the whole exceeds the sum of its parts (Elster 1989b: 133). In the following Elster's chain of collective action will be discussed.

Elster (1989b: 133-134) elaborates on how the interplay of individuals with different motivations may foster cooperation in social dilemmas. He assumes that there are three types of individuals with non-selfish behavior. These types are the Kantians, the utilitarians, and individuals motivated by the norm of fairness. Whether or not this classification of motivation is exhaustive the interplay between the motivations is a very interesting example.

Kantians act according to the Kant's (1995: 215) categorical imperative that states:[497]

> *… handle nur nach derjenigen Maxime, durch die du zugleich wollen kannst, dass sie ein allgemeines Gesetz werde.*

This means that one should only act according to a maxim that one wants to become a universal law. Kant's categorical imperative corresponds roughly to the question „but what if everyone did it?" (Elster 1989b: 56). Actual behavior according to the categorical imperative may be donating blood or voting for educational expenditures even one has no children (Arrow 1972: 349). The categorical imperative is not concerned with actual outcomes or what happens if an individual makes a decision. It is irrelevant whether a single action changes an undesired situation. The categorical imperative is concerned about the hypothetical outcome if everybody takes an action or everybody makes a donation. Similarly, the circumstances of the decision are irrelevant. The maxims are not contingent on what others do. If one thinks that according to a universal law nobody should steal it is irrelevant whether i) stealing is lucrative or not (actual outcome) and whether ii) all

[496] Some minor parts of the Java technology are not owned by Sun but by third parties. These third parties refused or partly refused to release their source code under the GNU GPL. Such encumbered source code is in consequence not released under the GNU GPL.

[497] Within the book "Groundwork of the metaphysic of morals" there are different formulations of the categorical imperative (Kant 1995, see pages 215, 216, 226, 232, 235, and 237). However, for the purpose of this study the differences do not seem to be relevant.

others steal or not (circumstances). Additionally, Kantians do not consider the costs of cooperation (Elster 1989a: 192).[498]

Utilitarians are concerned about actual outcomes and circumstances (Elster 1989b: 56). Utilitarian are sensitive concerning the consequences of their actions. If they donate money they give less the more other give because of the decreasing utility of money and vice versa.

Individuals motivated by the norm of fairness or reciprocity are concerned about circumstances but not about outcome (Elster 1989b: 56-57). They cooperate only if a substantial number of others cooperate as well (Elster 1989a: 187). Such individuals look what others do and follow some sort majority. If corrupt practices are standard the own behavior is defended by the claim “everybody does it”. For some of these individuals it is enough that half of the others cooperate as well. For other individuals a higher threshold of cooperation is necessary. Individuals concerned about the norm of fairness are not concerned about the outcome of their action. They donate money if everybody does independent whether the targeted goal is already reached or not.

Elster (1989a: 205; 1989b: 133) assumes that there are few Kantians, some utilitarians, and many individuals motivated by the norm of fairness.[499] In a possible scenario Elster demonstrates the dynamics of collective action based on a mix of these motivation types. He assumes that the first individuals to start collective action are the Kantians (Elster 1989a: 204).[500] They do not care that the outcome of their activity may be wasted because nobody else joins the cause. Kantians are also not concerned about the circumstance that other individuals free ride at this point. They just follow an universal law that the cause is a good one and one should therefore cooperate. At the very beginning of cooperation the utilitarians do not invest scare resources because they are not sure whether the collective good will be produced or not. Utilitarian individuals do not want to waste their resources. Individuals motivated by the norm of fairness do not cooperate at the beginning because the majority does neither as well. However, by cooperating unconditionally the Kantians make it more likely that the collective good is produced. Thereby the Kantians act as catalysts for the utilitarian. The utilitarian cooperate in consequence as well. Afterwards, individuals motivated by the norm of fairness start cooperating because others do. Each additional cooperator will trigger off other individuals motivated by the norm of fairness with a higher threshold. At the same time utilitarians reduce their cooperation because the collective good will be produced even without their contribution. This may lead either to a stable level of cooperation or to a collapse of cooperation because the individuals motivated by the norm of fairness withdraw as a reaction their cooperation as well. Anyway, the Kantians do not change their behavior because it is independent of outcomes and circumstances.

The example of Elster shows clearly that the distribution of motivation in a group may be very important for collective action (see also Fehr & Gintis 2007). If there are no Kantians in a group it is likely that nobody will start contributing to a collective good. For utilitarians success is not yet likely enough and individuals motivated by the norm of fairness refrain from contribution because others do as well. In a world of

[498] Elster (1989a: 193-194) cites the example of the Protestant pastor André Trocmé and other inhabitants of the village Le Chambon who did not consider the costs of behavior. They provided asylum for German Jews at great personal risk during the Second World War. They explicitly refused to consider the consequences of their action and relied on the principle to never turn away anyone who needs help.

[499] For a related empirical overview see chapter 5.4.1.

[500] Elster’s argument about Kantians is more complex (Elster 1989a: 192-193). However, the basic version of the story serves the purpose.

utilitarians or individuals motivated by the norm of fairness cooperation could hardly arise. Interestingly, even the tit for tat strategy starts with an *unconditional* first cooperative move. However, although Kantians are very important to start cooperation it is not necessary that all individuals are Kantians. If they give an impulse the chain reaction of cooperation may be kept alive by differently motivated individuals. Too many Kantians may even be a problem if the optimal level of cooperation is below universal cooperation (Elster 1989a: 194).

Elster's chain of collective action framed differently

Elster's argument about the importance of differently motivated individuals can be framed in another way to show its relevance for collective action: Fundamentalists start contributing to a collective good. They do not know whether their action will be successful and whether other individuals will join the cause. They just follow their moral obligation. Other individuals recognize that something is going on and that the struggle may be successful. Attracted by future rewards rational egoists contribute as well. This urges individuals with a preference for fairness to contribute their fair share as well.

The argumentation presented above does hold true to overcome the spiral of silence discussed briefly in chapter 6.2.2. Again, special individuals are needed who act as a seed for change. Changing society is in consequence reserved "to those who either know no fear of isolation or have overcome it" (Noelle-Neumann 1993: 139). John Locke for example was such a deviant character. He claimed that men are hardly concerned with God's commandments or the law of state. At that time it was possible to face capital punishment for such statements.

In the terminology of Max Weber (1990: 12) on social action an individual following *goal-instrumental* (zweckrational) action does not start building a public good because there is a low probability of reaching the goal. Similarly, an individual following *traditional* behavior (traditional) does not start acting because doing so is not based on past tradition. However, a *value-rational* (wertrational) person does start contributing to a public good irrespectively of success only because it is a valued goal. The same holds true for a person guided by *affectional* action (affektuell). If it is fun to contribute such a person contributes indeed. With the time going on success gets more probable which makes *goal-instrumental* action possible. Finally, contributing to the public good gets a custom leading to *traditional* behavior.

Mixed motivation and group heterogeneity may be beneficial

Elinor Ostrom (2000a: 138) notes that a central finding of research on collective action is that "the world contains multiple types of individuals" of which some are "more willing than others to initiate reciprocity to achieve the benefits of collective action".[501] This assumption is in contrast to assumptions in standard game theory and economic theory in which it is mostly assumed that preferences are homogenous (Erlei 2003: 2).[502] However, Fehr and Schmidt (1999: 856) conclude that a main insight of their analysis on game theory is "that there is an important interaction between the distribution of preferences in a given population and the strategic environment". In some environments a minority of purely selfish people may force the majority of

[501] As economic psychology tells us it may be possible that different motivational attitudes in different phases are the result of a change in attitudes within individuals (Lea, Tarpy, & Webley 1987: 11-13). The advertising business is heavily based on such changes.

[502] For exceptions see for example Fehr and Schmidt (1999) or Fischbacher and Gächter (2006).

fair-minded people to behave completely selfish as well. In games with punishment, however, a small minority of fair-minded individuals can force a majority of selfish player to fully cooperate.

In a computer simulation on social dilemmas Kondo (1990) found that cooperation can be maintained by rational egoists as long as the shadow of future is big enough. However, this cooperation is not stable and disturbances or deviant behavior may lead to a collapse of cooperation. If there are some individuals who behave normatively (in this case conforming to other's expectations) cooperation is stabilized. If there are additionally actors acting morally respectively unconditionally cooperative non-cooperation can be turned into cooperation. Based on this simulation it can be concluded that individuals who behave normatively are stabilizers while individuals who behave morally are catalyzers of cooperation.

As shown above mixed motivation of individuals may be beneficial for a successful production of collective goods. Tightly related to mixed motivation is group heterogeneity. Hardin (1982: 67) questions the usual assumption

- that groups are symmetric,
- that all group members have identical interest in the collective good,
- that all group members place the same value on a given unit of the collective good supplied, and
- that all group members place the same value on a cost unit.

For Hardin (1982: 67-89) the key for successful collective action is the existence of a subgroup in which members are willing to provide public goods. In heterogeneous groups a few individuals diverge from the average insofar as they contribute to a collective good while the big majority does nothing. They thereby solve the start-up problem. Woodworth (1918: 70) argues that voluntary contributions are needed in order to get a task started, to overcome repugnance, and to master inertia. Fulk (2004: 583) similarly argues that in the case of an organizational information common, attaining early commitment in the absence of clear benefits is particularly critical in establishing such a system. It demonstrates benefits to potential later adopters. Environmentalists care more about air pollution than ordinary citizens do. In consequence, they invest more and earlier scare resources for preserving clear air. Thereby these environmentalists act as catalyzers.

Peddibhotla and Subramanirgue (2007: 340) argue similarly that for public document repositories such as Wikipedia having a critical mass of individuals contributing at the beginning is very crucial. They found eight different types of motivations to contribute to public document repositories (Peddibhotla & Subramani 2007: 339). Basically, the authors found that motives linked to other individuals such as social affiliation, altruism, and reciprocity were negatively associated with the quantity of contributions but positively associated with the quality of contributions (Peddibhotla & Subramani 2007: 341). Motives oriented toward the individual itself such as self-expression, personal development, utilitarian motives, and enjoyment were positively associated with the quantity of contributions but negatively associated with the quality of contributions. The type of motivation was therefore clearly related to the quality and quantity of contributions.

In an interesting study of Fischbacher and Gächter (2006) the authors find not only that there is a considerable degree of heterogeneity in people's contribution preferences (2006: 13) but also that the interaction of individuals with heterogeneous preferences explained a large part of free riding dynamics (2006: 26). Similarly, for Erlei (2003: 27) the inclusion of heterogeneous preferences plays an important role in explaining the deviations of laboratory behavior form standard game predictions. The discussed assumption that preferences, motivations, and groups are heterogeneous is in contrast to assumptions in standard game theory

(Udehn 1993). However, as shown above heterogeneous motivation may be a crucial determinant for the successful production of collective goods.

Hardin (1982: 68; see also Olson 1995: 29) argues that other things equal asymmetric groups have a greater chance for successful collective action than symmetric groups (see also Santos, Santos, & Pacheco 2008). Based on findings of Arazy et al. (2006) on the quality of Wikipedia articles Ball (2007) argues that “the main lesson for tapping effectively into the 'wisdom of crowds', then, is that the crowd should be diverse”.

Lessons

Based on the discussion above on the interplay of different motivation it can be concluded that an analysis of collective action should consider the following insight:

i. Collective goods are seldom produced at once. In most cases contributions are dispersed within a certain period. Within this time frame there are different production phases in which the likelihood of success, structures, mechanisms, and circumstances may change. It is a different decision problem to spend the very first dollar for a hospital building in a developing country than spending the very last dollar. Incremental production of collective goods and a sequential model of decision making seem therefore to be an important aspect for explaining collective action (Heckathorn 1996: 272).

ii. Motivation in groups is distributed heterogeneously. During the different production phases of a collective good individuals face a change of structures, mechanisms, and circumstances. Not all types of motivation respond similarly to a certain set of structures, mechanisms, and circumstances. If there is a mix of different types of motivation prospective contributors may step in to contribute in all production phases. It may even be that successful collective action is *only* possible if group members have different types of motivation.

It can be summed up that different individuals have different motivations, that during the different production phases of a collective good not all motivation are similarly suitable, and that group heterogeneity may be beneficial for a successful production of collective goods. Unfortunately, the importance of different types of motivation is usually not taken into account in research on collective action (Peddibhotla & Subramani 2007: 330).

6.4 Excursion: Monday demonstrations in Leipzig 1989

The Socialist Unity Party of Germany (SED) of the German Democratic Republic celebrated on October 7, 1989 its 40th anniversary. The general secretary of the SED Erich Honecker called the state foundation a historical necessity. A few weeks later Honecker resigned, the Berlin Wall fell, and one year later the German Democratic Republic ceased to exist. Many western observers were stunned how fast the German Democratic Republic collapsed (Hirschman 1993: 173; Lohmann 1994: 43).[503] Commenting the collapse of the Eastern

[503] Not only the collapse of the German Democratic Republic but most major uprisings had been a very surprise (Kuran 1991: 44). In Poland the result of the parliamentary elections in 1989 (the trade union Solidarity won 260 out of the 261 eligible Senate and Assembly seats, the single seat captured by the ruling communist party was won by an atypical candidate who turned from a party member to a private entrepreneur (Castle & Taras 2002: 83)) was neither anticipated by opposition nor by the ruling regime (Kuran 1991: 10). The Russian Revolution in February 1917 was also widely unforeseen. Lenin told a gathering of youth socialists in Zurich (Switzerland) some weeks before the

Europe communist regimes the US funded propaganda radio station Radio Free Europe stated that "our jaws cannot drop any lower" (Kuran 1991: 7). The long-time representative[504] of the Federal Republic of Germany to the German Democratic Republic Hans-Otto Bräutigam said that in 1988 he would have responded to somebody who was to state that the German Democratic Republic will join the Federal Republic of Germany in 1990 with the reply that this prediction is "nonsense" (Gysi 1999: 140).

A prominent factor contributing to the fast breakdown of the German Democratic Republic were the mass protests in Leipzig known as the Monday demonstrations.[505] The Monday demonstrations in Leipzig started at September 4, 1989 and ended at December 18, 1989.[506] It was the first large scale protest since 1953.

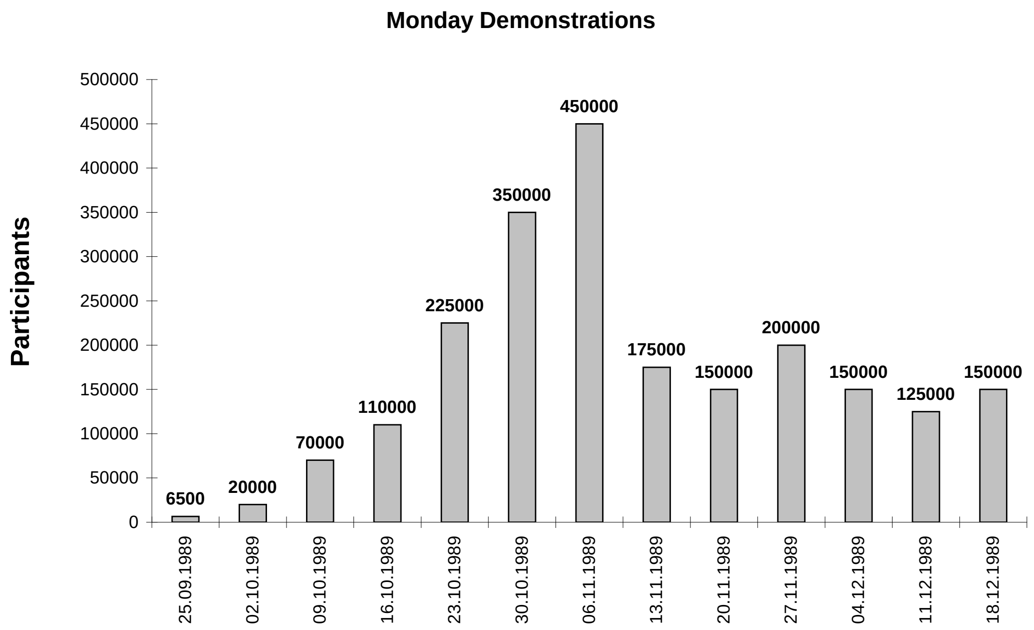


Figure 79: Development of participants in the Monday demonstrations in Leipzig[507]

February Revolution that Europe will not escape social upheaval but that "we old-timers perhaps shall not live {to see} the decisive battles of the looming revolution" (Pipes 1996: 112). A few days before the revolution started foreign observers reported to their capitals that the Tsarist capital Petrograd is stable and secure (Chamberlin 1935: 73-76). The fast and unforeseen destruction of order can also be seen in the incidences after the landfall of Hurricane Katrina near the city New Orleans on 29. August 2005. Very quickly the destruction of order manifested in looting as well as shooting at rescue helicopters and relief workers. Only a few days after the incident Governor Kathleen Blanco was forced to warn lawbreakers quite harshly that the National Guard troops are "hardened on the battlefield in Iraq" and "know how to shoot and kill" (Breed 2005). Other examples for surprisingly successful revolutions were the French Revolution in 1789 and the Iranian Revolution. In all these cases neither the CIA nor the KGB, neither Louis XVI nor Robespierre, neither the Shah nor Ayatollah Khomeini, and neither Tsar Nicholas II of Russia nor Lenin anticipated the revolutionary success.

[504] Because of its Basic Law the Federal Republic of Germany did formally never accredited the German Democratic Republic. Bräutigam was therefore officially not called ambassador. However, he was de facto ambassador of the Federal Republic of Germany to the German Democratic Republic.

[505] Next to these demonstrations for political liberalization, open borders, and German unification there were four other demonstration cycles (Lohmann 1994: 42-43). The succeeding cycles are less known and their purposes were different from the first demonstration cycle. The following discussion refers only to the first cycle.

[506] In fact, there were demonstrations and protest since the formation of the German Democratic Republic and since 1982 there were each Monday prayers for peace in the Nikolai Church in Leipzig (Opp 1991: 302). However, the number of involved individuals was quite small.

The number of participants in the demonstrations showed an exceptional pattern. In Figure 79 the development of participants in the Monday demonstrations in Leipzig is presented. Until the peak at November 6 the number of demonstrators increased sharply. Afterwards this quantity fell and started to oscillate. There are three main questions associated with this pattern and the upheaval:

i. Why was the upheaval such a surprise to everyone?

ii. Why did the movement attract uniformly more adherents and gain momentum *until* November 6?

iii. Why was there a sharp decline in demonstrators *after* November 6?

In the following, these three questions will be discussed to draw afterwards some general conclusions for this dissertation.

6.4.1 Why was the upheaval such a surprise to everyone?

To answer why the upheaval was such a surprise to everyone the question of timing of upheavals must be discussed. Gurr (1970) prominently suggests that it is hardship which is the mother of revolt.[508] According to this argumentation the timing of the upheaval must be explained by the reasoning, that the situation got very fast so bad that people started to revolt. In consequence, there must have been an unanticipated event that worsened the situation of the citizens dramatically. However, Elster (1989b: 17-18) argues that this unlikely true. People badly off have a high motivation to rebel but their capacity to do so is the lowest when they are in tight circumstances.[509/510] Tocqueville (1856) noticed similarly that revolution do not occur in the worst off region of a continent or a country. In 1789 feudal suppression was higher in Germany than in France. Similarly, the economic condition in Paris was better than in other parts of France. Nevertheless, the French Revolution started in France respectively in Paris. Tocqueville also observes that revolutions occur if the situation improves and not if it gets better. Similarly, the Soviet Union (SU) did not collapse during Stalinist harshness but during Gorbachev's glasnost that is associated with freedom of speech, management transparency, and openness of debates. Nye (2006) notes that "glasnost let people say what they thought" and in consequence "many people said, 'We want out'". Hardship is therefore perhaps necessary but not sufficient for revolutions. In consequence, the sudden collapse of the German Democratic Republic cannot completely be explained by a single event that worsened the situation.

Another explanation for the timing of the upheaval is that the demonstrators just followed the opposition. However, there were since 1982 each Monday prayers for peace in the Nikolai Church in Leipzig and there were demonstrations as well as protest since the formation of the German Democratic Republic (Opp 1991: 302). In fact, there was always an opposition in the German Democratic Republic. The existence of an opposition is in consequence not a sufficient trigger for an upheaval.

---

[507] Opp (1991: 303). For the demonstrations at September 9 and at September 25, 1989 there are no or no accurate estimations (Opp 1991: 303). Based on various sources Lohmann (1994: 70) reports slightly different numbers. However, the pattern with increasing participants until the peak at November 6, 1989 and afterwards a more or less stable number of participants remains very similar.

[508] For an overview over theories that attempt to explain the nature and origins of revolutions see Goldstone (1980).

[509] Applying the hardship argument to business one may argue that necessity is the mother of innovation (Elster 1989b: 17-18). However, innovation requires resources and investment that near bankrupt firms in need for innovation may hardly afford.

[510] Sen (1997: 1) argues that „a perceived sense of inequity is a common ingredient of rebellion". However, he further states that the possibility to rebel matters.

The explanation of Kuran (1991: 43) is that there was a latent bandwagon in the German Democratic Republic. There were always citizens protesting and showing their discontent (for a multifaceted overview see Poppe, Eckert, & Kowalczuk 1995). This was nothing new. However, this protest was not big enough to induce a change. This may have prompted bystanders to conclude that the protests will never have any effect. A series of events then lowered the threshold of bystanders and abutted the bandwagon. These events were:

- The success of the opposition in the semi free elections in Poland in 1989 (Kuran 1991: 10): The trade union Solidarity won 260 out of the 261 eligible Senate and Assembly seats. This showed that opposition in a communist state might be successful.
- The announcement of Soviet Foreign Affairs spokesman Gennadi Gerasimov that the SU has shifted from the Brezhnev Doctrine to the Sinatra Doctrine (Jones 2005a: 80): Under the Brezhnev Doctrine the internal affairs of satellite states were tightly controlled by the SU.[511] According to this doctrine an upheaval would have been defeated by Soviet troops. In fact, this happened a few times during the Cold War. The Sinatra Doctrine, a hint to Frank Sinatra's song "{I did it} my way", stated that the SU does not interfere in internal affairs of other states even if they break with communism. The opposition could therefore assume that Soviet troops do not defeat an upheaval.
- The mass migration from the German Democratic Republic to Austria via Czechoslovakia and Hungary (Hirschman 1993: 187): This signaled a serious decline in state authority and increased the courage of demonstrators. In the words of Hirschman (1993: 177) exit (migration) and voice (demonstration) worked in tandem and reinforced each other (see also Pfaff 2006; Pfaff & Kim 2003). However, exit played the lead part in the collapse (Hirschman 1993: 187).

The migration to Austria, the decision of the SU not to interfere in East Germany, and the success of the trade union Solidarity in Poland boosted the opposition enormously. These events gave the latent bandwagon a gentle push. This resulted in consequence in a bigger size of the opposition that in turn induced that several citizens reached their threshold and so forth (see also chapter 6.4.2). The bandwagon quickly took on speed.

The distribution of thresholds in a non-free political system is not known. Because everybody is scared to speak out his grievances one does not know whether other citizens are against the regime as well. Although there were always surveys in the former communist states they either were perceived as irrelevant or actively suppressed (Henn 1998: 232). For a long time the instruments by which public opinion was discovered included information provided by the secret police, the party apparatus, letters to the mass media, and self-criticism. Similarly, it was thought that party cadres know about the citizen's needs, attitudes, and preferences. In the late 80ies opinion research started to get more important (Henn 1998: 234-235). However, many people were afraid of revealing any opinion that might incriminate them (1998: 238). For example in Poland interviews were often associated with government officials, the police, security services, and the communist

[511] At the Fifth Congress on the Polish United Workers Party on November, 12 1968 Leonid Brezhnev stated: „When external and internal forces hostile to socialism try to turn the development of a given socialist country in the direction of the restoration of the capitalist system, when a threat arises to the cause of socialism in that country, ... this is no longer merely a problem for that country's people, but a common problem, the concern of all socialist countries" (Ouimet 2003: 67).

party. It was not believed that the surveys were anonymous (1998: 240). In Germany only right before the reunification surveys seemed to have gained momentum.[512]

In chapter 5.6 it was argued that the stability of social order that is based on external forces and positive and negative selective incentives is low. If the incentives are gone social order collapses immediately. Additionally, in chapter 6.3 it was shown that the punishment of defectors might have a big impact on individual decision making. The punishment in communist dictatorship was not only based on state security forces but also on the big majority of ordinary citizens. Havel (1985: 37-39) suggests persuasively that in the communist regimes of that day it cannot be drawn a line between victims and supporters of such sanctions.[513] The line between victims and supporters of suppression has to be drawn through each person. Each person is in his own way victim and supporter; of course with different degrees of involvement. Havel points this up with the story of a greengrocer who does not put a label with the slogan "Workers of the world, unite!" in his shop window. Consequently, he will be relieved of his post as a shop manager, his pay will be reduced, he will not be able to make holiday in Bulgaria, his children will not get access to higher education, and his superior will start harassing him. Most of these sanctions are not applied by state security but by regular citizens. They will not do so out of a true inner passion but simply because they are scared of sanctions. They do not want get victims themselves. In this sense individuals are both victims and supporters of sanctions and punishment. It could be argued that the small events described above drove regular citizens to stop monitoring and sanctioning other citizens. Thereby the system of mutual monitoring and sanctioning disposed. In consequence, the old social order collapsed very fast.

It can be summarized that that the “Wende” in Germany was a surprise because *firstly* people were accustomed that protest does not have an effect. It was thought that the bandwagon always remains latent. The migration to Austria, the decision of the SU not to interfere in East Germany, and the success of the trade union Solidarity in Poland changed the situation slightly. Thereby the threshold of a few additional citizens was reached. They joined the opposition and lowered thereby the threshold of other citizens and so forth. Consequently a chain reaction started. *Secondly*, the distribution of preferences and protest thresholds were not known. It was therefore not anticipated that only a small seed could foster almost unanimous opposition. *Thirdly*, stability was hold up by regular citizens. At the moment they refused to monitor and sanction others the old social order collapsed instantly.

6.4.2 Why did the movement gain momentum uniformly until November 6?

According to Opp (1991: 305-306) the decision to oppose a regime actively depends next to dissatisfaction on *how likely an individual believes* that:

i. this opposition is instrumental for the production of the desired public good,

---

[512] For a list of surveys conducted in the German Democratic Republic see the datasets provided by the Central Archive for Empirical Research (ZA 2001). However, during parts of the history of the German Democratic Republic surveys were secret and only produced for the “safe” (Gysi 1999: 161).

[513] Interestingly, Solézenicyn (2005: 160) argues similarly: „If only there were evil people somewhere insidiously committing evil deeds, and it were necessary only to separate them from the rest of us and destroy them. But the line dividing good and evil cuts through the heart of every human being. And who is willing to destroy a piece of his own heart? During the life of any heart this line keeps changing place; sometimes it is squeezed one way by exuberant evil and sometimes it shifts to allow enough space for good to flourish. One and the same human being is, at various ages, under various circumstances, a totally different human being. At times he is close to being a devil, at times to sainthood.“

ii. the opposition will only be successful if everybody contribute to it (self-efficacy),

iii. he will be punished, and

iv. the opposition is thought to be successful.

In the following, these factors will be discussed for the Monday demonstrations. Thereby it will be explained why the upheaval gained uniformly momentum until November 6.

Opposition as an instrument for the public good

Participants of the demonstrations and the German Democratic Republic population differed in their political opinion (Lohmann 1994: 77-78). A study conducted between the 20th and 27th November 1989 showed that one third of the population were member of the SED, that more than half of the population intended to give the SED their vote in free, democratic, and anonymous elections, and that only one-third of the population supported unification with Germany (DJI 1989a: 10-11 and 20; for methodological problems see Henn 1998). A study about the participants of the Monday demonstrations at December 4, 1989 showed a different picture.[514] Three quarters supported unification, 6% intended to give the SED their vote in free, democratic, and anonymous elections, and 6% were member of the SED (DJI 1989b: 10, 12, and 29). It can be assumed that for the demonstrators a unification and the abolishment of the one party system[515] was considered a good cause.

Whether the demonstrators assumed the protest instrumental for the production of the public good “political reforms/exit possibility” or not cannot be corroborated. Nevertheless, people knew how effective protest could be because of the example of Poland. Some months before the Monday demonstrations started the protest of Solidarity lead to semi free elections in which the trade union defeated the communist party completely. It can therefore be guessed that people thought that protest might be instrumental for reaching their goals.

However, even if protest was thought to be instrumental the participation of individuals in the protests can still not be completely explained. It is for individuals in their self-interest to let others make the sacrifices that are needed for a regime’s downfall (Kuran 1991: 14). Because everybody enjoys the benefits of a downfall demonstrations and protests are public goods.

Self-efficacy

As already discussed in chapter 5.4.2.1.1 self-efficacy is an important variable for inducing contribution to collective goods. Some special individuals like the two pastors at the St. Nicholas Church Friedrich Magirius and Christian Führer could have believed that their own contribution were crucial for success. However, the

---

[514] However, the above cited preferences of the demonstrators did not seem to be sharp contrast to the whole population. In fact, an analysis of the secret police concluded that the regular population does basically not differ from the citizens who try to escape from the German Democratic Republic (ZAIG 1989b). Furthermore, the secret police found that even long time party members were very critical to the regime and that they did not differ in this regard from the regular population (ZAIG 1989a). It can therefore be argued that the demonstrators did probably not differ from the population in the preferences concerning the downfall of the regime but in their propensity to tell their true opinion and to raise their voice. Participating in the Monday demonstrations signaled clearly the own opinion. So it made for demonstrators not much sense lying about the own opinion. For the general population this was very different.

[515] In fact, there were several other parties besides the SED pooled in the National Front. However, they had neither any influence nor were they independent from the SED.

regular demonstrator could hardly assume that he or she is crucial for success. Whether 450’000 or 450’001 demonstrate does not make any difference. Self-efficacy could therefore reasonably not be an important factor in the Monday demonstrations.

Punishment

At least until October 19 the sanctions demonstrators had to face were substantial. This could include prison, destruction of the professional career, or exclusion from education for the whole family (Opp 1991: 310). Not only were sanctions severe but also likely. Demonstrations were generally terminated immediately and the protestors arrested. Sievers (1990: 85) reports for example that at the Monday Demonstration of September 11 the demonstrators were closed in and many of them arrested. The demonstrators knew about the massacre at the Tiananmen Square in Beijing just a few months ago. The German Democratic Republic showed publicly unconditional solidarity with the Chinese communist regime indicating that it will respond similarly. There were some serious rumors of an imminent “Chinese solution” in October 1989 (Hirschman 1993: 193) and in fact the Chairman of the Council of State Honecker signed the order to shoot at demonstrators (Lohmann 1994: 69).[516] In addition, credible stories circulated that hospital floors in Leipzig were emptied and supplies of frozen blood readied in anticipation of a flood of new arrivals (see for example Kallenbach & Schade 1998). The army was put on standby to support the police of Leipzig and issued their soldiers twice as much ammunition as normal. In the most read newspaper an article asked to stop the counterrevolutionary actions immediately; if necessary with weapons. Additionally, some of the older demonstrators probably have experienced the violent suppression of the 1953 mass uprising with the help of Soviet military forces that caused many fatalities.[517] Not until October 1989 Jochaim Gauck (2000: 49) could say “Wir sagen unserer Angst ‚Auf Wiedersehen’ “(We say goodbye to our fears) as he explained in a now renowned speech.[518]

Yin (1998: 535) concludes that “an important feature of collective rebellion is people deciding to join based on knowing how many others have already participated”. With increasing size of a crowd the anticipated costs for acting get lower (Berk 1974: 366). One reason for this is that the probability of being caught decreases (Yin 1998: 536). In fact, with an increasing number of individuals joining the Monday demonstrations the probability as well as severity of sanctions sunk. From October 19 on security forces stopped disintegrating demonstrations.

The factors of Opp (1991) refer to external costs and benefits. They depend mainly on the regime reaction and the size of public opposition (Kuran 1991: 16-25). However, Kuran suggests including also internal costs and benefits. Internal costs are based on the desire to “live within the truth” as Havel (1985: 39) impressively called it (see also Wagnerová 2006)[519]. If this desire is not satisfied a loss of personal autonomy and a sacrifice

[516] Who finally ordered not to shoot despite Honecker’s command is still not clear. There are many different individuals who claim to be responsible for this refusal to obey orders.

[517] Several demonstrators were sentenced to death by courts, killed by tanks, or died by gunshots. Diedrich (1991: 293-296) estimates that 54 persons died. The BStU (2007) reports that between 60 and 80 demonstrators and between 10 and 15 policemen and soldiers died. The BStU estimates that between 13’000 and 15’000 demonstrators were arrested.

[518] Joachim Gauck is known as the first “Federal Commissioner for the Records of the National Security Service of the Former German Democratic Republic”.

[519] Shortly after having written this in October 1978 Vaclav Havel was put to prison until 1983.

of personal integrity must be accepted.[520] Therefore, the costs of punishment must be subtracted from the psychological benefits to live within the truth that results in a net benefit or net loss.

Success

Success could reasonably not be expected at the very beginning of the Monday demonstrations. Each Monday since September 1982 there were prayers for peace at the St. Nichols Church in Leipzig without changing the situation. Through the increasing opposition size the perceived influence grew as well. With time going on the disaggregation of the regime was clearly observable; for example by the forced resign of the Chairman of the Council of State Erich Honecker at the October 18. An aspect of success is power in number (Yin 1998: 536). Only a big crowd is capable to conduct efficient actions against a regime (Berk 1974: 363). This crowd grew continuously indicating increasing probability of success. Roughly three-quarters of the participants of the Monday Demonstration at December 4, 1989 expected that the renewal of society will succeed (DJI 1989b: 6).[521]

Merging the factors

The different factors can now be summed up. By the Polish example the potential demonstrator could see that protesting may be instrumental in the dissolution of the regime. While the number of demonstrators increased the costs of punishment decreased. At some point these costs were smaller than the benefits to live within the truth (see also the discussion of low-cost situations in chapter 6.1.1). With every other protestor joining the cause the threshold for participation of another individual was reached and so forth. At the same time the probability of success grew constantly.

At September 25 a very small group of 6500 individuals demonstrated in Leipzig. They accepted very high costs of punishments and a low probability of success. These protestors had most probably a very strong and above the average desire for personal freedom. Because of this size of demonstrators the participation threshold of other 13’500 individuals was reached. They joined the cause forming a group of 20’000 demonstrators. The chain reaction went on. After October 30 one could assume that the security forces did not punish the demonstrators anymore. With a crowd of 350’000 demonstrators it seemed to be obvious that the state lost the possibility to punish the participants individually. Thereby other 100’000 protestors showed up the next Monday.

The question why the demonstrations attracted so many adherents and why it gained momentum until November 6 can be partly answered:

i. Some citizens had an enormous inner desire to oppose the regime. This desire was enough high to accept substantial sanctions while success was unlikely. In Granovetter’s (1978) term they had a very low threshold. These people showed their opposition each Monday in Leipzig for almost a decade. Their opposition constituted a latent bandwagon (Kuran 1991: 43) and these persons served as catalyzers.

[520] Kuran (1989: 47) does include the utility an individual receives from a certain social order. The influence of a regular individual on the social order is negligible which is also obvious to this individual. The social order and the associated utility are therefore given.

[521] It was not specified how the renewal should look like.

ii. Because of several exogenous factors the costs for demonstrations were lowered and success got more probable. The Sinatra Doctrine for example made clear that the Soviet troops will not dissolve demonstrations. This was a very important information because the involvement of Soviet troops would have been dangerous for the demonstrators. The successful migration of citizens to the west showed how vulnerable the regime was which made success more likely. In consequence a few additional supporters of the opposition emerged which otherwise perhaps would not have. The Sinatra Doctrine as well as the migration to Austria gave the bandwagon a little nudge.

iii. With more demonstrators joining the cause the probability of success increased and the probability and severity of sanctions decreased. The threshold of many additional potential supporters was reached which resulted in further participants. The bandwagon gained momentum and a chain reaction started.

iv. With a big crowed already protesting the probability of success and sanctions were shifted again. Even citizens with a high threshold started to participate. Thereby the bandwagon gained full speed.[522] At this point it would have been difficult to stop the opposition.

6.4.3 Why was there a sharp decline in demonstrators after November 6?

The sharp decline in the number of demonstrators after November 6 is due to an exogenous event. On November 9 Günter Schabowski announced, after a misunderstanding concerning dates, in a live broadcasted international press conference that all traveling restrictions were abolished *immediately*.[523] A part of the demonstrators changed their form of opposition from voice to exit (Hirschman 1970). The citizens who wanted to migrate to the Federal Republic of Germany reached their goal and stopped demonstrating. Others intended to change the political system and kept demonstrations going on. The objectives of the demonstrators did therefore not remain constant (Lohmann 1994: 86). This can also be recognized by looking at the texts of the chants sung at demonstrations. At the very beginning the chanting was „Wir wollen raus!" (We want to get out) from people who wanted to leave the German Democratic Republic (Schade 2003: 43). With the time going on other demonstrators who wanted to change the political system joined the demonstration and started to chant "Wir bleiben hier!" (We stay here). After some competition between these two fractions they merged under the slogan "Wir sind das Volk!" (We are the people) and finally "Wir sind ein Volk" (we are one nation, Hirschman 1993: 190; Schade 2003: 43).

As argued above the preferences of the demonstrators were not homogenous. Hirschman (1993: 176) states that in 1989 there were different goals related to collective action. However, there were not only different goals but also different means of action. At the very beginning of the Monday demonstrations exit and voice increased their efficiency mutually (see chapter 6.4.1). However, after November 6 exit respectively the

[522] Interestingly, not only the number of participants within the German Democratic Republic increased faster and faster but also the speed of the opposition in the different communist countries. A banner in Prague stated "Poland – 10 years, Hungary – 10 months, East Germany – 10 weeks, Czechoslovakia – 10 days" which Kuran (1991: 42) complemented with "Romania – 10 hours" (see also Bikhchandani, Hirshleifer, & Welch 1992: 993). While in Poland the opposition, namely Solidarity, struggled for years for freedom the Romanian upheaval was very fast.

[523] It was already possible before to file an application to migrate to the west. Yet migration was not always accepted and in either case associated with serious drawbacks (Opp 1991). Citizens asking for migration to the west had to anticipate repressive sanctions. If the application was luckily accepted all social contacts had to be abandoned and no assets could be taken along.

migration to the Federal Republic of Germany impaired the power of voice. To be effective voice often requires group action and is therefore subject to free riding. Exit in contrast does not need any coordination with others. Hirschman suggests therefore that exit tends to undermine voice. This is particularly the case when exit deprives the potential carriers of voice of their most articulate and most influential members (Hirschman 1993: 176). In case of the Monday demonstrations exit reduced the power of voice. However, this was no big problem because the opposition had almost entirely succeeded.

6.4.4 What can be learned?

From the Monday demonstrations four major findings are interesting for the advancement of this dissertation:

i. The distribution of individual thresholds was not known what caused that everybody was surprised by the success of the upheaval.

ii. The demonstrators had in respect to their participation different thresholds. This explains the gradual increase in the number of protestors until November 6. Some individuals accepted to bear substantial costs to advance their cause. Others agreed with these individuals concerning the cause but did not want to undergo the high costs at the beginning.

iii. Not all demonstrators had the same preferences or goals („Wir wollen raus!" versus "Wir bleiben hier!"). This observation is crucial for explaining the decrease in the number of protestors after November 6.

iv. For starting protest individuals with a very strong inner conviction were needed. The very first protestors accepted to bear substantial costs that can only be balanced by very strong preferences for a change. These protestors had a huge desire to live within the truth. Thereby these protestors acted as catalyzers and solved the start-up problem.

These findings lead to the following lessons:

i. To understand the dynamics of collective action it may be necessary to pay special attention to differently motivated individuals.

ii. Even if individuals are similarly motivated it must be acknowledged that these individuals may have different thresholds.

iii. For social movements it is crucial to solve the start-up problem.

6.5 Summing up

The basic insights of this chapter are:

- Individuals do not act in isolation but in interdependence with other individuals. > See chapter 6
- For certain collective goods a critical mass of contributions is needed before it is beneficial at all. The reasons for this are *firstly* that the payoffs associated with contributions to collective goods are highly dependent on the amount of others who have already contributed. *Secondly*, the behavior of other may serve as an information repository that may guide the own behavior. > See chapter 6.1

- Unlike in the prisoner's dilemma the costs for cooperation in social dilemmas may vary contingent on the cooperation rate. After a tipping point respectively a threshold is reached in the production of a collective good it may be beneficial to contribute even for rational egoists. The distribution of thresholds in a society influences the dynamics of collective action. > See chapter 6.2
- The dynamics of collective action depends heavily on incremental decision making. Herding behavior and bandwagon effects are based on what others have done before. > See chapter 6.2
- Solving the start-up problem and reaching a critical mass are essential for bandwagon goods. After the start-up problem is solved collective action may be self-sustaining. A possibility to solve the start-up problem for electric appliances is to have a stand-alone application that is valuable without network effects. > See chapter 6.2.3
- Bandwagon effects are also observable in FOSS development. Above all reputation investors depend on bandwagon effects because they need an audience. > See chapter 6.2.3
- Different individuals have different types of motivation. In the production of collective goods these motivations are differently served depending on the development phase. > See chapter 6.3
- It is argued that a mix of motivation as well group heterogeneity is very important for explaining collective action. In collective action individuals who behave morally may be catalyzers, individuals who behave normally may be stabilizers, and rational egoists burden a large amount of the contributions. > See chapter 6.3
- The Monday demonstrations in Leipzig 1989 supports the assumption that in collective action i) not all individuals have the same preferences, that ii) not all individuals have the same threshold, and that iii) to start a social movement and to overcome the start-up problem individuals with a very strong motivation are needed. A latent bandwagon remained immobile until the Sinatra Doctrine and the success of Solidarity in Poland gave the latent bandwagon a gentle push. This resulted in consequence in a bigger size of the opposition that in turn induced that several citizens reached their threshold and so forth. Each individual joining actively the opposition lowered the costs for participation of others and increased the likelihood of success. The bandwagon quickly took on speed. > See chapter 6.4

In this chapter two very important steps towards hypotheses generation are made. *Firstly*, the importance of the dynamics in collective action is shown. *Secondly*, motivation is connected to this dynamic respectively to different phases in the production of collective goods. It is also highlighted that different types of motivation do not have the same importance in different production phases. This suggests that governance structures and mechanisms must be adapted to the production phase of a collective good. Individuals with different types of motivation are differently related to certain governance structures and mechanisms.

7 Organizational life cycles and group dynamics: Drive in organizations and groups

In the previous chapters sequential decision making (see chapter 5.4.2.1.1) and different phases in the production of collective goods (see chapter 6.3) were discussed. Such dynamics have not only been discussed in social dilemma literature but also in research on organizational life cycles and group dynamics. In the following, this literature is briefly discussed.

7.1 Organizational life cycle

Organizations are continually changing (Child & Kieser 1981: 28). This may be due to external conditions such as competition, innovation, public demand, or governmental policy. However, organizations must not only cope with its environment but also with internal challenges (Cangelosi & Dill 1965). It is for example possible that the composition of the workforce demands an organization to develop. Organizational development is additionally not only a function of strategy but also of organizational ageing. A successful organization that grew from a small enterprise to a large-scale enterprise will hardly be able to use the same problem solving processes as after start-up. Although the business is the same the mere size of the organization will demand changes. At the same time there may be furthermore some degeneration associated with organizational development. In such situations managers knowing the old days say “why don’t we have the spirit of excitement we used to have?” (Lippitt & Schmidt 1967: 102). Some authors (e.g. James 1974) suggested therefore comparing the development of organizations with living organisms by applying a life cycle of organizational development. According to such life-cycle concepts organizations go trough the phases of emergence, growth, maturity, decline, and degeneration or regeneration[524] (James 1974: 49-54) respectively through the phases of birth, youth, and maturity (Lippitt & Schmidt 1967).

It was heavily discussed how life cycles influence organizational development (for a good overview see Child & Kieser 1981). James (1974: 49-54) researched the impact of the life-cycle concept on finance, marketing, production, and administration. He argued for example that administration in the emergent phase is very loosely defined. During the growth phase the organization will strengthen control systems and begin to apply formal personal policies to regulate the activities of the growing number of employees. In the maturity phase administration depends heavily on systems and adherence to formalized control. Administrative patterns itself start being more important than the purpose they were designed for. The moral of the employees decreases while the personal turnover rate increases. In a regeneration path reappraisals of corporate needs are made while in a degeneration path skillful employees leave and it gets difficult to replace them.

In Greiner’s (1998) model an organization progress through five phases:

[524] Because obviously some firms manage to survive (Child & Kieser 1981: 46) some authors distinguished for the last stage between a degeneration or a regeneration path (James 1974). The analogy to a living organism ends thereby.

- The ***first phase*** is dominated by ***creativity***. The founders are technically or entrepreneurially oriented. Communication among employees is frequent and informal. Long hours of work are rewarded by modest salaries and the promise of ownership benefits. Decisions and motivation are highly sensitive to marketplace feedback and the management reacts to customers. With the time going on a crisis of leadership emerges. Due to informal procedures it is unclear who actually leads the organization.
- The ***second phase*** is dominated by the struggling for an ***appropriate direction***. A functional structure is introduced and job assignments become increasingly specialized. Accounting systems for inventory and purchasing are introduced. Incentives, budgets, and work standards are adopted. Communication becomes more formal and impersonal. At the same time a hierarchy of titles and positions grows. Lower-level supervisors are treated as functional specialists and not anymore as autonomous decision makers. With the time going on a crisis of autonomy emerges. Lower-level employees find themselves restricted by the stern hierarchy.
- The ***third phase*** is dominated by ***delegation***. Plant managers and market territories are given more responsibility. Profit centers and bonuses are introduced to motivate employees. The top management often concentrates on acquiring enterprises. Communication from the top management is infrequent. With the time going on a crisis of control emerges. Autonomous managers run their own shows without coordinating plans, money, and technology with the rest of the organization.
- The ***fourth phase*** is dominated by ***coordination***. Decentralized units are merged into product groups. Each product group is treated as an investment center. Formal planning procedures are established and reviewed. Certain technical functions such as data processing are centralized while daily decisions remain decentralized. Stock options are used to encourage employees to identify with the entire organization. With the time going on a red tape crisis emerges. Managers learn to justify their action to a watchdog audience at the headquarter. A lack of confidence between line and staff respectively between field and headquarter builds up.
- The ***fifth phase*** is dominated by ***collaboration***. It is characterized by the focus on solving problems quickly through team action. Key managers frequently discuss major problems in conferences. Teams are combined across functions to handle specific tasks. Economic rewards are geared toward the team and not the individual. A matrix-type structure is frequently used. Staff at headquarters is reduced in number and the formal control system is simplified. With the time going on a crisis around psychological saturation of employees emerges. Employees get emotionally and physically exhausted from the intense teamwork.[525]

In a review of nine models of organizational life-cycles (Adizes 1979; Downs 1967; Greiner 1972; Katz & Kahn 1978; Kimberly 1979; Lippitt & Schmidt 1967; Lyden 1975; Scott 1971; Torbert 1974)[526] Quinn and Cameron (1983: 34-41) found some similarities.[527] In all models organizations progress through the following life-cycle phases:

- The ***entrepreneurial phase*** is characterized by marshalling resources, lots of ideas, entrepreneurial activities, little planning and coordination, formation of a niche, and powerful prime movers.

---

[525] Funnily, the model stops here although a possible crisis is envisioned.
[526] For an updated reprint of the Greiner article from 1972 see Greiner (1998).
[527] For another review as well as an interesting empirical methodology see Miller and Friesen (1983b; 1984).

- The ***collectivity phase*** is characterized by informal communication and structure, a sense of collectivity, long hours spent, a sense of mission, continued innovation, and high commitment.
- The ***formalization and control phase*** is characterized by formalization of rules, stable structures, emphasis on efficiency and maintenance, conservatism, and institutionalized procedures.
- The ***elaboration of structure phase*** is characterized by an elaboration of the structure, decentralization, domain expansion, adaptation, and renewal.

Newer contributions adopt organizational life-cycles to small professional service firms (Masurel & van Montfort 2006), New Zealand wineries (Beverland & Lockshin 2001), management accounting systems (Moores & Yuen 2001), human resource management problems (Rutherford, Buller, & McMullen 2003), cooperative interorganizational relationships (Jap & Anderson 2007), and stakeholder theory (Jawahar & McLaughlin 2001). However, models of organizational life cycles have had its heydays and little conceptual progress can be observed lately. Nevertheless, the main message seems to be still valid: Governance structures and mechanisms change and must change during different development phases of organizations.

7.2 Group dynamics

The idea that groups go through certain phases has a long history (Bales & Strodtbeck 1960: 624; for a comparison of different models see Heinen & Jacobson 1976: 101-102).[528] Bales and Strodtbeck (1960: 624) define phases as different subperiods in continuous group interaction to solve a problem. A group has to make decisions and proceeds thereby from initiation to completion of the problem. Based on this definition it is then clear that there is no sharp but a fluent transition from one phase to the other. A group member is therefore not always able to clearly specify in which phase he is. However, most probably this individual knows whether a group is in its initiation or completion phase.

The most prominent concept of group dynamics is the one of Tuckman (1965).[529] Based on a review of literature dealing with group development he induced that there are the following phases (for a short description see also Schermerhorn, Hunt, & Osborn 2005: 198-200; Tuckman 1965: 396):

- ***Forming***: Groups are concerned with orientation by identifying the boundaries of interpersonal and task behavior. Relationships and dependencies to leaders, other group members, or preexisting standards are established.
- ***Storming***: The second phase is characterized by conflict and polarization around interpersonal issues. Emotions influence the task sphere considerably.
- ***Norming***: Ingroup feelings and cohesiveness develop, novel standards evolve, and new roles are adopted.
- ***Performing***: Group roles become flexible and group energy is channeled into the task. Structural issues are solved and the group structure is now supportive concerning task performance.

Later Tuckman and Jensen (1977) reviewed another 22 studies that had appeared since the original study (see also Tuckman 1984). They added a fifth phase called adjourning. In this phase the task is completed and the

---

[528] For an interesting article on group dynamics see Cartwright and Zander (1960).

[529] Tuckman's (1984: 14) concept is an example of an influential paper that was first rejected by the scientific community.

group breaks up. Interestingly, groups with a short group life such as laboratory groups (a few hours) as well as long lasting therapy groups (a year) followed the same patterns just described.

A reason why Tuckman's concept was so successful is perhaps that it withstands the test of common sense.[530] Many individuals who have worked in groups could probably report that at the beginning group member show an orienting behavior. People are actually quite polite to each other. However, if the newness has worn off the group members react emotionally. Yet, this is followed by pulling together. Finally, in the ideal case the group becomes a functional instrument for dealing with the task.

Another important concept of group phases has been developed by Bales and Strodbeck (see also Bales 1950; 1960). They argue that there are three phases of group dynamics:

- In the first phase the interaction in groups is related to problems of ***orientation*** (e.g. ask or give information, clarification, or repetition). It is decided what the situation is like.
- In a second phase problems of ***evaluation*** dominate (e.g. ask or give opinion or analysis). It is reflected what should be done.
- In the final phase problems of ***control*** take over (e.g. ask or give suggestion or direction). The group actually decides what to do.

During these phases ***negative reactions*** (e.g. show tensions, withhold help, show antagonism, deflate other's status) as well as ***positive reactions*** (e.g. show tension release, give help, show solidarity, joke, laugh) increase. Related to these assumptions Bales and Strodtbeck set up the predictions presented in Figure 80.

Predicted relationships between group problems and group phases

		Group phases	
Importance of Problems	*First*	*Middle*	*Final*
Orientation (1)	High	Intermediate	Low
Evaluation (2)	Low	High	Intermediate
Control (3)	Low	Intermediate	High
Negative reactions (4)	Low	Intermediate	High
Positive reactions (5)	Low	Intermediate	High

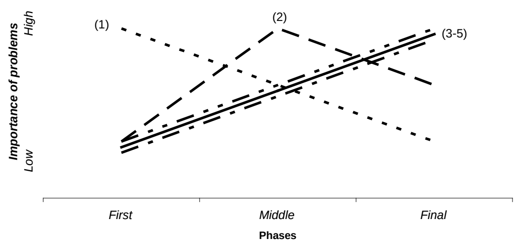


Figure 80: Predicted importance of problems in different phases[531]

Bales and Strodbeck (1960, for a methodological discussion see Bales 1950) tested their assumptions about group dynamics in 22 different group sessions. To do so they related the individual interactions to orientation, evaluation, control, negative reactions, and positive reactions. In fact, if the data of all group sessions is

[530] Tuckman (1984: 14) argues that another reason for the success of the concept may be the wording "forming-storming-norming-performing".

[531] Based on Bales and Strodtbeck (1960: 634).

summed up the reactions during the different group phases corresponds exactly with the hypothesis (see Figure 81).[532] Interactions related to orientation decrease during the different group phases. The peak of evaluation related interactions is found during the middle phase. Control related interactions as well as negative and positive reactions increase from the first to the final phase. It can be concluded that group dynamics depends on the phase a group runs through.

Summed relationships between group problems and group phases in 22 groups

Importance of Problems	*Group phases*		
	First	*Middle*	*Final*
Orientation (1)	1668	1170	916
Evaluation (2)	1550	1792	1656
Control (3)	285	364	429
Negative reactions (4)	275	374	408
Positive reactions (5)	984	1058	1361

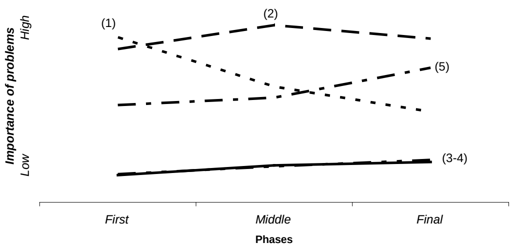


Figure 81: Importance of problems in different phases[533]

7.3 Summing up

The basic insights of this chapter are:

- During their life cycle organizations change their structures and mechanisms. The first organizational phase is dominated by creativity, technical orientation, and autonomy. In consequence, a leadership crisis emerges and structures get more formalized. This causes some individuals to fell restricted by hierarchy. As a result tasks are delegated that in turn causes coordination problems. They are solved by formalized planning. The members of an organization learn to justify their action to a watchdog audience at the headquarter. Finally, teamwork is given more importance. > See chapter 7.1
- Group dynamics are characterized by forming phase that is followed by a storming, a norming, and finally a performing phase. At the beginning questions of orientation are prominent. With the time going on orientation problems decrease while problems of control increase. > See chapter 7.2

This short chapter has the following aim: It shows that i) dynamical forces and ii) development phases are not only important in social dilemmas (see chapter 6) but also in organizations and groups. One learns that governance structures and mechanisms have to be adapted to the related challenges. The subchapter provides also inspiration for i) the methodology of this dissertation and ii) for hypotheses generation concerning individuals motivated by peer recognition.

[532] If the data is not aggregated and the single groups are observed the pattern of problem importance corresponds in some cases with the hypothesis while in others the pattern is different (Bales & Strodtbeck 1960: 632).

[533] Bales and Strodtbeck (1960: 635).

8 The dynamics of collective action in FOSS projects: The basic idea and the hypotheses

Motivation is a major factor influencing collective action (see chapter 5.4.1). At the same time it was shown in this study that situational factors (see chapter 5.4.2) are very important for explaining collective action. Particularly decision sequence was presented as a significant factor influencing collective action. It was further argued that motivation interacts with these situational factors in general but also with group phases respectively organizational life cycles (see chapter 7). In chapter 5.6 it was concluded that motivational as well as structural provisions are needed to solve the problems of collective action.

Motivation is also a very important factor in FOSS development (see chapter 3.3). It seems to be clear that different motivational forces are important for explaining contributions to FOSS projects. Situational factors are central for FOSS development too (see chapter 4.2 and 4.3). Furthermore, that situational factors foster or inhibit subjects with diverse types of motivation differently seems to be very plausible. Situational factors are related to FOSS development phases. These development phases provide in turn distinct incentives, opportunities, and limitations for contribution to FOSS. Governance structures and mechanisms change during these development phases. In consequence it is basically hypothesized that in FOSS development i) different types of motivation, ii) governance structures and mechanisms, and iii) development phases interact with each other and that this interaction is crucial for the success of FOSS.

In scientific literature the influence of motivation on participation in FOSS projects has already been discussed by several studies. However, the influence of motivation contingent on governance structures respectively mechanisms and development phases has not been discussed at all. It is thought that this is a major research gap. In the following it is suggested *how* motivation, governance structures respectively mechanisms, and especially development phases interact.

Basically, for explaining collective action in FOSS it is suggested to adopt a dynamic view. It is argued that research on FOSS should acknowledge

i. that different types of motivation respectively heterogeneous motivation are important,
ii. that governance structures and mechanisms are important,
iii. that development phases are important, and
iv. that the above mentioned factors interact with each other.

This chapter starts by presenting the basic assumptions about the dynamics of collective action in FOSS projects. This includes the discussion of the production function of FOSS, incremental decision making, and code reuse. Afterwards, the hypotheses to be tested are presented. Finally, the differences of the model adopted here and threshold models are mentioned.

8.1 Basic assumptions

Collective action problem

A basic assumption of this study is that FOSS is a public good (see chapter 2.9.2). Therefore, a collective action problem (see chapter 5.2) has to be solved.

It was argued that the costs respectively benefits of cooperation may vary contingent on the cooperation rate (see chapter 6). After a tipping point respectively a threshold is reached in the production of a collective good it may be even for rational egoists beneficial to contribute. In consequence, a bandwagon may start to roll if the initial start-up problem is solved. Corresponding effects are also observable in FOSS development (see chapter 6.2.3). In consequence, it is assumed that the characteristics of collective action in FOSS development change contingent on the development phase.

In FOSS development different individuals have different motivations. In the production process of collective goods these motivations are differently served depending on the development phase. Some individuals with a special motivation contribute at an early phase. Such individuals can hardly be motivated by the outcome of their action. The first few contributions will unlikely change anything (see chapter 5.7). These individuals can also hardly be motivated by the circumstances such as for example reciprocal fairness. Reciprocal fairness requires that have already contributed. However, with development going the situation changes. Individuals jump on the bandwagon because they consider now success likely. Circumstances (e.g. everybody else contributes) or outcome (e.g. application can be used for production) may get important. Prominently, reputation investors depend on a bandwagon already in motion. Investors in reputation need an audience (see chapter 3.2.2.3).

Based on the argumentation presented above it must be stated that it is wrong to believe that collective action generally and FOSS development specially can always be explained by one type of motivation (see chapter 5.6 and chapter 6.3). Individuals with different motivation contribute in different development phases of a collective good. In consequence, different motivations are complementary in solving social dilemmas.

Production function

It was argued that the collective action problem in FOSS projects is related to development phases. Therefore, the corresponding production functions (see chapter 5.7) is crucial.

There are two assumptions concerning the development of the value of software applications:

- *Firstly*, there are important network externalities and switching costs for FOSS applications (see chapter 2.9.3). An application just having implemented basic functionalities does hardly have many users yet. In consequence, there are hardly any positive network externalities. For example the value of certain software applications increase if the files can be exchanged with other users. Similarly, if there are few other users of a certain application one cannot get help easily. For switching costs an argumentation as for network externalities can be applied. For FOSS applications switching costs are high in the initial development phase. In an intermediate phase the number of users increase. An application that cannot attract more users will hardly keep momentum and survive the initial phase. By having more users switching costs decrease and positive network externalities increase. However, at some point most relevant individuals use the software. At that point there is almost certainly somebody around to help. Additional users do therefore not increase the value of the software eminently. Therefore, based on network externalities and switching costs the value of software increases slowly in the initial phase, increases exceptionally in the intermediate phase, and increases slowly in the mature phase.

- *Secondly*, it needs some effort until a stable version of software runs with minimal functionality (see however software reuse below). Until that point the value of the good is very low. Software that crashes, does not run, or compute wrong result cannot be exerted for productive use. An operating system that does not properly control and allocate memory, prioritize system requests, control input and output devices, facilitate networking, and manage file systems is not of much value. With development going on the software matures and the first stable and runnable version has gone gold.[534] Minimal functionalities are implemented and that is why the value of the software increases considerably. Every new functionality adds something important. Thereby the relationship between input (e.g. time) and output (e.g. improved functionality) is beneficial. However, at some point the most basic functionalities are implemented. It needs then an extra effort to have value added.

Based on the arguments stated above the value of most FOSS projects does not increase linearly (see Figure 82).[535] In an initial phase the value of software remains low despite much effort has already been executed. After some point of development the value of a FOSS application increases very rapidly. However, this momentum in an intermediate phase cannot be maintained forever. The value still increases in a mature phase but the advancement gets lower. The production of software seems therefore to be somehow S-shaped (see chapter 5.7 and Figure 82).

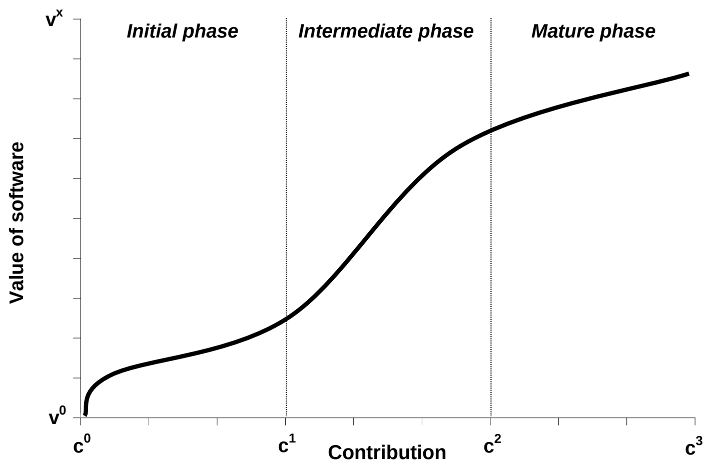


Figure 82: FOSS production function

[534] Software that has gone gold is ready for regular users. The term “gone gold” refers to software ready for mass production. A gold release is the final version of a software application. The term “gold” seems to refer to the usage of gold leaf for the master record in phonograph production (Edison 1892). Using gold leaf was revolutionary for the mass production of phonographs (Gracyk & Hoffmann 2000: 14).

[535] The value of a FOSS project refers to the value for potential users apart from the developers. While software that crashes permanently is not of much value for a user this must not be necessarily the case for a developer. The value for a developer may be high because he learns a lot by investigating system crashes. Increasing human capital does not completely depend on whether the software can be used for production.

Figuratively, in the initial phase there is a latent bandwagon. Somebody has to give this latent bandwagon a nudge. It needs the efforts of especially motivated individuals to produce some initial source code and initial functionalities. After a critical mass is reached the bandwagon gains speed in the intermediate phase. After this point is reached self-interested developers may consider it worthwhile to jump on the bandwagon. Every additional developer joining the cause increases the value of the software. In consequence the threshold for others to contribute is reached. Thereby the speed of the bandwagon increases. At some point the bandwagon is at full speed that allows covering a large distance in relative short time. However, at some point the value added by additional contributors gets smaller. In consequence, the speed of the bandwagon in the mature phase decreases.

S-shaped production function: An example

In the example presented in Figure 83 the production function of a software project is visualized. Not much value is added until the first stable version 1.0 is released.[536] The succeeding major version 2.0 provides new features and therefore much value. However, the value of the software application does not increase smoothly from version 1.0 to 2.0. Security and small bug fixes included in Version 1.1 provide much value with relative low effort. The alpha release of the new version 2.0 will not contribute much value to a user. The same holds true for the beta version. Both can hardly be used for production and will crash frequently. However, the step from the release candidate (rc1) to the final version 2.0 adds much value to a user with relative low effort. This release candidate should not have major bugs that need to be repaired for the final version. For version 3.0 more effort has to be employed to increase the value of the software application. The most useful features have already been implemented in Version. 1.0 and 2.0.

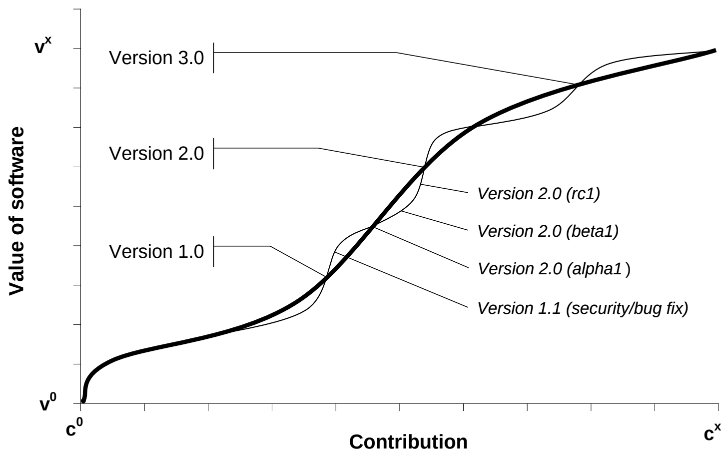


Figure 83: Generic software production function

Production function is not universal

Certainly, the described pattern is not universal. One may find software projects in which the production function is different. There are surely software applications for which there no major switching costs or

[536] Whether the first runnable and stable version is called 0.1, 1.0, or 3.582 is irrelevant for the argument.

network externalities. In addition, for some applications it may be easy to implement all needed functionalities with low effort. However, it is thought that for many FOSS projects an S-shaped production function is true.

The main problem

If the production function is S-shaped a main problem is to solve the start-up problem (see chapter 5.7 and 6.3). According to Behlendorf (1999: 162) the "essential to the health of an open-source project is that the project have sufficient momentum to be able to evolve." Berglund and Priestly (2001: 134) argue similarly that a FOSS project must generate a critical mass of developers who contribute such as happened for Linux or Apache.

Incremental decision making

In chapter 2.8 it was shown that software is developed incrementally. At the same it was argued in chapter 5.4.2.1.1 that incremental decision making has important consequences for collective action:

- *Firstly,* incremental decision making is accompanied by informational asymmetries. Being second to move eliminates certain outcomes and strategies. Information asymmetries give an advantage to subsequent players. Rational egoists can maximize their welfare by focusing on the best response contingent on the decision of precedent movers. This line of reasoning can also be adopted to FOSS development. If software developers contribute to an already successful project they can be quite sure that their effort is not wasted. A reputation investor contributing to a FOSS project that is already heavily used by commercial firms may assume that there is a possibility that he can produce a valuable signal. Individuals who care about procedural fairness know whether other individuals cooperated or not. In consequence, they may respond reciprocally.
- *Secondly,* incremental decision making is also accompanied by commitment asymmetries. Moving first is in fact a strong commitment because it cannot be reversed. By commitment to a sequence of actions a player may be able to alter the play of potential contributors to a public good. This does also apply to FOSS projects. At the beginning of a project the first contributors have to make a decision about the license. A very credible commitment for other FOSS developers is to choose an appropriate license (see chapter 2.6.2). Having chosen the GNU GPL makes sure that the contributed source code remains free because this licensing act can legally not be reverted. For pro-socially motivated developers such a commitment is very important. If a project is licensed under the GNU GPL they can be sure that their contributions are in either case not lost to their cause and remain free software. The GNU GPL sets up a very strong protection mechanism (see chapter 2.6.2). Therefore, project managers can make a very credible commitment by choosing a certain FOSS license. Having chosen a license is not only important in legal respect. The license determines the context (see chapter 5.4.2.1.2) of the project. It tells basically whether it is okay to keep source code private or not and serves also as a sort of constitution (see chapter 2.6.2).

Therefore, the dynamics of collective action depends heavily on incremental decision making (see chapter 6). The contribution of some types of programmers is influenced by informational asymmetries. Such individuals base their contribution on the knowledge that others have contributed as well and that there is some likelihood for success. Incremental decision making allows early contributors also to make a credible commitment respectively choosing an appropriate license. If for example a project is licensed under the GNU GPL pro-social developers can be sure that that the project remains free. Therefore, some developers may be influenced by commitment asymmetries.

Incremental decision making and different types of motivation

In an initial phase fun seekers contribute unconditionally to a project because they enjoy producing source code. They do not care whether success is likely or not or whether the software can be used safely on production machines or not. This is very beneficial because it starts cooperation. In the terminology of Elster fun seekers are the catalyzers of FOSS development (see chapter 6.3). If developers motivated by peer recognition join fun seekers they may trigger reputation investors, investors in human capital, and pro-socially motivated developers to contribute as well. The value of the software is monotonically increasing[537] and strategic interaction or selective incentives get possible solutions for the social dilemma. Afterwards, the application matures and developers interested in adjusting it to personal requirements contribute as well. At some point a large proportion of the public good is produced and success is foreseeable. Developers interested in the success of FOSS application will reduce their efforts.

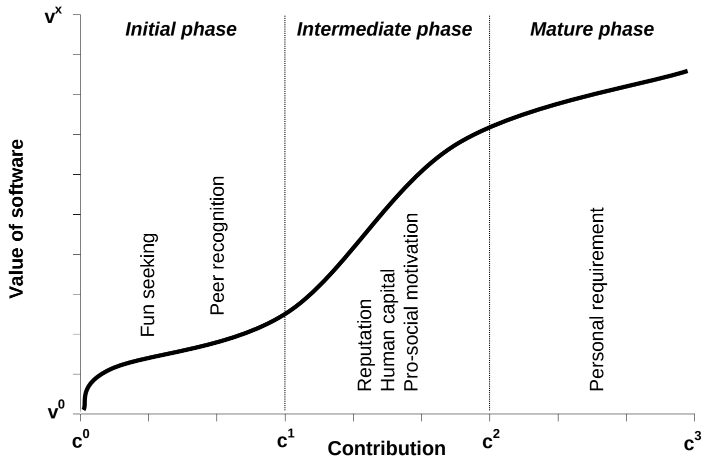


Figure 84: Generic development phases in FOSS projects

The argumentation presented here supports the assumption that not only incremental decision making is important for explaining FOSS development. It must be further acknowledged that developers have different types of motivation and that group heterogeneity may be important.

Continuous decision making

Related to incremental decision making is continuous decision making (see chapter 5.4.2.1.1). Continuous contribution facilitates *firstly* to start cooperative in uncertain situations. With a small contribution it can be found out whether others reciprocate. In nobody reciprocates not much is lost. By far more important for FOSS development is *secondly* that actors whose motivation is small can make a small contribution. In FOSS

[537] For a mathematical description of increasing functions see Jeffreys and Jeffreys (2000: 22).

development a programmer motivated by a personal requirement may contribute only a small bug fix. If there are many such bug fixers contributing small pieces of source code development may be fortified considerably.

Software reuse

In FOSS development the start-up problem is somewhat reduced but not solved by the reuse of source code (see chapter 4.3.8). It is assumed that depending on the topic 20% to 85% of source code can be reused. By source code reuse it is possible to produce *relatively* fast a runnable version. If somebody for example wants to create a new web browser he may rely on an existing layout engine (e.g. Gecko). By doing so a developer does not have to develop source code that deals with formatting information (e.g. CSS) and displaying it on the screen. In consequence, a runnable prototype or an alpha version of the web browser is *relatively* fast ready for preliminary testing.

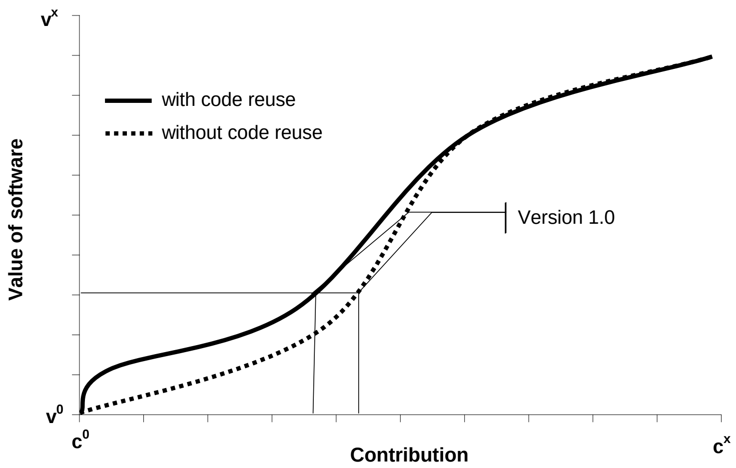


Figure 85: Generic software production function with and without source code reuse

8.2 Hypotheses

After having discussed the basic assumptions above the precise hypotheses are now developed below. However, first of all some preliminary remarks are made and the structure of the argumentation is presented.

Preliminary remarks

There are two kinds of hypothesis: alternative and null hypotheses. The researcher believes that the alternative hypothesis is true. However, because a verification of the alternative hypotheses is too troublesome he tries to reject the null hypotheses.[538] A null hypothesis is set up so that is supports an alternative hypothesis if it can be rejected. To disprove a hypothesis is easier than to prove it because only one single failure is needed. In contrast, a prove must test all cases. Fisher (1971: 16) argued that the null hypothesis “is never proved or established, but is possibly disproved, in the course of experimentation. Every experiment may be said to exist only in order to give the facts a chance of disproving the null hypothesis”.

[538] For the rationale of falsification see Popper (1984: 14-17 and 47-59).

However, the alternative hypotheses are not proved if the null hypotheses are rejected but they are more likely than before the rejection (Popper 1984: 198). With any attempt to falsify a theory it is more and more corroborated.

For convenience, the following hypotheses are presented as alternative hypotheses.

Structure of the hypotheses development

In a first step, hypotheses concerning the motivation of FOSS developers are presented. Afterwards, hypotheses related to motivation *and* governance structures respectively mechanisms are discussed. In a further hypothesis the interaction between motivation and governances structures respectively mechanisms is connected to satisfaction with participation in FOSS development. Afterwards, hypotheses concerning motivation and the type of engagement are mapped out. A few hypotheses deal with motivation and community affiliation. Finally, the main hypotheses related to motivation and development phases round the chapter off.

8.2.1 Hypotheses about motivation

In chapter 3 the motivation of developers to contribute to FOSS projects was discussed. Empirical evidence suggests that fun seeking and investment in human capital are the two most important types of motivation. Pro-social motivation, personal requirement, and peer recognition are less important. Investment in reputation seems to be the least important motivation. In consequence, it is hypothesized that fun seeking and investment in human capital are the two most important types of motivation and that investment in reputation is the least important motivation.[539]

Hypothesis 1: *Fun seeking and investment in human capital are the two most important types of motivation.*

Hypothesis 2: *Reputation investment is the least important motivation.*

In chapter 3.2.2.3 personal requirement respectively scratching an itch as a motivation to contribute to FOSS development was discussed with some skepticism. It is usually argued that an itch is scratched because the developer needs a functionality implemented or a bug fixed to use the software for production. However, it was suggested that doing so is costly and that in economic terms the next-best solution “buying new hardware”, “buying new software”, “accept the bug”, or “accept inadequate functionality” is in many cases more beneficial. It was further argued that a developer might also scratch an itch because he is intrinsically interested in solving the problem. If so he may be wiling to spend a considerable amount of scare resources such as time without gaining much added value. Solving a technical problem does therefore not only serve a personal requirement but is also fun. However, it is thought that many developers equate scratching an itch with personal requirement although their real motivation is to have fun. Therefore, it is thought that personal requirement is related to fun seeking. The motivation personal requirement should therefore be positively correlated to fun seeking. The same argumentation also applies to investment in human capital. It is thought that scratching an itch respectively personal requirement is related to investment in human capital.

[539] Because the data of this study is based on a non-probability sample (see also chapter 9.3 and footnote 574) these two hypotheses cannot be answered completely reliable.

Hypothesis 3: *a) Personal requirement is positively related to fun seeking.*

b) Personal requirement is positively related to investment in human capital.

The different types of motivation were characterized in chapter 3.1 by reference to self-determination theory. Based on the degree of self-determination and the perceived locus of causality the motivations were put on a continuum from intrinsic regulation (fun seeking), to integrated/identified regulation (pro-social behavior), to introjected regulation (peer recognition), and finally to external regulation (personal requirement, reputation, and human capital). Therefore, it seems to be reasonable to assume that intrinsic regulation, integrated/identified regulation, and introjected regulation group while motivation related to external regulation group as well.

Hypothesis 4: *Fun seeking, pro-social motivation, and peer recognition group while personal requirement, personal requirement, investment in reputation, and investment in human capital group as well.*

8.2.2 Hypotheses about motivation *and* governance structures respectively mechanisms

As discussed earlier in this study the likelihood of collective action is connected to the decision situation (see chapter 5). One aspect of the decision situation in FOSS projects, namely the governance structures and mechanisms, has been presented in chapter 4. Based on the incentives provided by these governance structures and mechanisms it is hypothesized how the different types of motivation (see chapter 3) are affected.

8.2.2.1 Assignment for project positions

It was argued that individuals in social dilemmas prefer not having a leader (see chapter 5.4.2.1.2). Only if the production of the collective goods suffers from coordination problems individuals require to install a leader. It is assumed that there is only a small coordination problem if there are only few developers. Therefore, it is assumed that very small projects with few developers have no special mechanisms for decision making.

Hypothesis 5: *The fraction of projects that have no special mechanisms for assigning project positions is considerably lower for very small projects.*

In chapter 4.3.7 two different generic decision making processes in FOSS projects were discussed: Hierarchical benevolent dictators who rule top-down and democratic bottom-up decision making. In the case of the benevolent dictator special positions in the project (e.g. release manager or project leader) are appointed. If decision making is democratic project positions are either assigned by voting or informal consent.

It was argued that intrinsic motivation flourishes if there is high autonomy (see chapter 3.2.1). It can be assumed that informal consent is the least and the benevolent dictator the most severe restriction to autonomy. Consequently, it can be argued that developers motivated by fun seeking choose more frequently projects in which decisions are made by informal consent.

Hypothesis 6: *Fun seeking is positively related to projects in which project positions are assigned by informal consent.*

The motivation of being recognized by peers is based on interaction and communication (see chapter 3.2.2.2). This should also hold true for decision making processes. Informal consent and voting respectively elections involve interaction and communication. Appointment by a leader does include the community only to a lesser extent. It can therefore be concluded that individuals motivated by peer recognition will avoid projects in which project positions are assigned by appointment.

Hypothesis 7: *Peer recognition is negatively related to projects in which project positions are assigned by appointment.*

Reputation investors strive to produce a valuable signal. The value of a signal depends partly on its visibility (see chapter 3.2.2.3). Being chosen as a maintainer by informal consent produces probably not a very visible signal *to outsiders*. This decision includes some fuzziness. In projects with such a decision making mechanism it is probably in some cases not clear who is the maintainer of a module. Being appointed is more visible because there is a clear binary decision. The same applies to cases in which maintainers are put in position by elections. Additionally, if somebody is elected the decision is made by many different developers and not by one or few developers as in the case of appointment. A decision based on many developers may look more reliable *to outsiders* than a decision based on one individual.[540] That a developer has been elected includes or is alleged to include also the information that the decision making was competitive. It can therefore be argued that for reputation investors being elected is the most preferred assignment procedure.

Hypothesis 8: *Reputation investment is positively related to projects in which positions are assigned by elections.*

It is thought that the assignment mechanisms are less important for individuals motivated by investment in human capital, by pro-social orientation, and by personal requirements.

8.2.2.2 Structures and mechanisms

In the following subchapter it is hypothesized how the different types of motivation are related to governance structures and mechanisms in FOSS projects (see also Franck & Jungwirth 2003a: 8).

Fun seeking

In chapter 3.2.1 it was argued that autonomy, competence, and relatedness foster intrinsic motivation. It can therefore be assumed that projects with a high degree of autonomy, competence, and relatedness are positively related to fun seeking. It was also argued that positive feedback enhances intrinsic motivation. In consequence, the degree of positive feedback should be positively related to fun seeking.

Hypothesis 9: *a) Fun seeking is positively related to autonomy.*

b) Fun seeking is positively related to competence.

c) Fun seeking is positively related to relatedness.

[540] A hacker rooted in the FOSS community may consider the decision of a highly reputed benevolent dictator more credible or reliable than the decision of a group of developers.

d) Fun seeking is positively related to positive feedback.

Fun seekers are not primarily interested in the outcome of their activity. Therefore, they are not expected to contribute above average to projects that would be missed by the users and the community if it were to disappear. However, fun seekers are interested in the activity itself. That is why they can be expected to engage in projects that are cool and would be therefore missed by the developers if they were to disappear.

Hypothesis 10: Fun seeking is positively related to projects that would be missed by developers if they were to disappear.

Pro-social motivation

Pro-social motivation is based on the values of the individuals (see chapter 3.2.2.1). Such values are reflected in the license terms that serve as a sort of constitution (see chapter 2.6.2). These terms specify what it is allowed and what not. It seems therefore to be clear that pro-social motivation is related to being happy with the license of a project because it serves as a constitution. An individual who cares about the social and political implications of FOSS would hardly contribute to a project with licenses terms that were to run completely against the own values.

Hypothesis 11: The intensity of pro-social motivation is positively related to happiness with the license of a project.

Having access to computer resources is very central to hacker ethic. Many FOSS developers think that software should not only *be* free but also that it should *remain* free. This is guaranteed by strong copyleft licenses (see chapter 2.5.1.2). Therefore, one may assume that pro-social motivation is positively related to projects with strong copyleft licenses (see also Franck & Jungwirth 2003a: 7 and 11).

Hypothesis 12: The intensity of pro-social motivation is positively related to projects with strong copyleft licenses.

Because the target of pro-social motivation is the well-being of others it can be assumed that pro-socially motivated individuals participate in projects that would be missed by its users and the entire community but not necessarily by the developers if they were to disappear.

Hypothesis 13: a) Pro-social motivation is positively related to projects that would be missed by the users if they were to disappear.

b) Pro-social motivation is positively related to projects that would be missed by the entire community if they were to disappear.

Peer recognition

Peer recognition is based on approval from others. It seems therefore reasonable to assume that peer recognition is related to relatedness (see chapter 3.2.1) in a FOSS project. The number of programmers or peers should also be relevant. A project with only one other developer will hardly attract an individual motivated by peer recognition and a desire to communicate with others.

Hypothesis 14: a) Peer recognition is positively related to relatedness.

b) Peer recognition is positively related to the number of programmers contributing to a project.

Individuals motivated by peer recognition engage most probably in projects that would be missed by the entire community and its developers but not necessarily by the users if they were to disappear.

Hypothesis 15: *a) Peer recognition is positively related to projects that would be missed by its developers if they were to disappear.*

b) Peer recognition is positively related to projects that would be missed by the entire community if they were to disappear.

Personal requirement

Contributors to FOSS projects may do so because they have a personal requirement. They should therefore clearly contribute to projects that would be missed by the users but not necessarily by the community or the developers if they were to disappear.

Hypothesis 16: *Personal requirement is positively related to projects that would be missed by the users if they were to disappear.*

Some regular users of FOSS applications motivated by a personal requirement have the know how and the incentive to install Linux software that it is not included in a major distribution such as Debian, Fedora, Ubuntu, Slackware, or openSUSE (see Footnote 37). However, to install software not included in a major distribution needs time. There may be dependencies between different software. In such cases time has to be invested to fix problems. Additionally, the user is responsible for updating the software. It is therefore easier to use Linux software already included in a major distribution. The contribution of individuals with a personal requirement should therefore be positively related to the inclusion of a project in a major Linux distribution.[541]

Hypothesis 17: *Investment in personal requirement is positively related to projects that are included in a major Linux distribution.*

Reputation investment

In chapter 3.2.2.3 it was argued that that the value of a signal is related i) to its visibility and ii) to its credibility. A project that would be missed by the entire FOSS community if it were to disappear is certainly a visible project. Furthermore, it can be assumed that a signal produced in a project missed by the entire FOSS community is credible. The same argumentation applies to a lesser extent to a project that would be missed by its users if it were to disappear. However, in a project that would be missed only by the developers a visible and credible signal can less likely be produced.

Hypothesis 18: *a) Reputation investment is positively related to the degree a project would be missed by the entire community if it were to disappear.*

---

[541] Certainly, there are important FOSS applications that do not run on Linux but on Windows, BSD, Macintosh, Solaris, HP-UX, or AIX. Therefore, this hypothesis does only apply to software that runs on Linux.

b) Reputation investment is positively related to the degree a project would be missed by the users if it were to disappear.

It was argued that there are two categories of signals in FOSS development (see chapter 3.2.2.3). One category of signals is based on *deliberately* created credit mediums such as credit files, websites, and the like. The characteristic of these signals is that they are easily accessible for outsiders , even for computer-dummies. The other category of signals, browsing source code, is not easily accessible for outsiders. Browsing source code and above all interpret it properly needs a considerable effort and, even more important, significant knowledge.

It can be assumed that a project that credits its contributors *deliberately* influences the decisions of reputation investors to contribute positively. If credits are *easily accessible by outsiders* more reputation investors should contribute than if not. Additionally, it can be assumed that the number of means to credit is also positively related to the propensity of reputation investors to contribute. If credits are given with different means the possibility increases that the search strategy of an employee is meet.

Hypothesis 19: *a) Reputation investment is positively related to the existence of deliberately created credit mechanisms.*

b) Reputation investment is positively related to the number of different means to credit contributions.

Related to signals is the involvement of commercial firms in FOSS projects (see chapter 3.2.2.3 and 3.4.1). Signals are not valuable if nobody is interested in them. It can be argued that projects in which commercial firms are involved are more interesting for commercial forms in general. Therefore, signals are more likely produced in projects with commercial firm involved because they are more likely valuable.

Hypothesis 20: *Reputation investment is positively related to the involvement of commercial firms.*

Some authors argued that visibility and credibility is related to the number of programmers contributing to a project. Therefore, the number of programmers should be positively correlated to investment in reputation.

Hypothesis 21: *Reputation investment is positively related to the number of developers in a project.*

Investment in human capital

One reasoning why contributions to FOSS projects enhance human capital is peer review (see chapter 3.2.2.2). It is thought that comments and feedback about contributed source code enables learning. Therefore, feedback should be positively related to investment in human capital. A similar relation should exist to the degree to which decisions in FOSS projects are explained. Only if decisions are explained there is the possibility to learn from them.

Hypothesis 22: *a) Investment in human capital is positively related to feedback giving.*

b) Investment in human capital is positively related to the degree decisions are explained.

The importance of competence respectively an *optimal* challenge was discussed for intrinsic motivation and fun seeking (see chapter 3.2.1). However, it seems to be reasonable to assume that individuals interested in enhancing their human capital are also interested in an *optimal* challenge. If the challenge is too low there is nothing to learn. If the challenge is too high complexity will inhibits the individual from learning any lesson.

Hypothesis 23: Investment in human capital is positively related to competence.

It can be assumed that individuals motivated by investments in their human capital choose to contribute to projects that are missed primarily by the developers but not necessarily by the community or the users. It is probably more appealing to learn from developer driven projects than by user driven projects.

Hypothesis 24: Investment in human capital is positively related to projects that would be missed by the developers if they were to disappear.

8.2.2.3 Paid developers and the involvement of firms

In chapter 3.4.2 possible problems between paid and non-paid developers were briefly discussed. Related hypotheses are discussed below.

Apparently, paid developers contribute to project that are appealing for commercial companies. Therefore, paid developers should contribute more likely to projects to which commercial firms contribute as well.

Hypothesis 25: Paid developers are more likely engaged in projects to which commercial firms contribute as well.

In chapter 3.2.1 and 5.6 the relationship between intrinsic motivation and externally implemented incentives was discussed. It was argued that intrinsic motivation is undermined by external intervention if individuals feel that their self-determination is adversely affected respectively if they consider these interventions to be controlling. Paying developers for their contribution is certainly an external intervention (see chapter 3.2.2.3). It may therefore be argued that being paid is negatively related to intrinsic motivation respectively fun seeking. Additionally, it is argued that the three types of externally regulated motivation, namely personal requirement, investment in human capital, and investment in reputation, are not related to the state of being paid. Motivation that is already externally regulated should not be influenced by an additional external regulation.

Hypothesis 26: a) Fun seeking is negatively related to the status of being paid.

b) The motivational types related to external regulation are not affected by the status of being paid.

At June 15, 2006 there were 122’520 registered projects at the hosting platform SourceForge (2006a). Surely, many of these projects will not attract many users. Therefore contributing to a project at its initial phase brings along uncertainty (see chapter 5.4.2.1.1). The most obvious but certainly not the only possibility to deal with this uncertainty is to wait with contribution until the intermediate phase. At this point of development it can be recognized whether there is some chance for success.[542] Linux is an example of a project to which commercial firms contributed only after success was predictable. The development of Linux started in 1991

[542] Success refers to the use of the software by independent users (see also footnote 535).

(see chapter 2.2) but not until the very late 1990s and early 2000s commercial firms jumped on the bandwagon and started to contribute. In chapter 5.4.2.1.1 it was argued that in social dilemmas deciding second has informational advantages. Therefore, it seems reasonable to assume that commercial firms do not contribute to FOSS projects in an early phase.[543] They wait until success is predictable.

Hypothesis 27: The development phase is positively related to the involvement of commercial firms.

If the involvement of commercial firms is positively related to the development phase and paid developers are more likely engaged in projects to which commercial firms contribute as well it can be concluded that paid developers are more likely engaged in projects that are in a later development phase.

Hypothesis 28: Paid developers are more likely engaged in projects that are in a later development phase.

8.2.3 Hypotheses about the interaction of motivation and governances structures/mechanisms

It was hypothesized that motivation is related to governance mechanisms and structures. It was argued that depending on the type of motivation particular governance mechanisms and structures foster motivation. In consequence, if the particular structures and mechanisms in a project are present an individual should be happier with his involvement in the project and vice versa (see also Hackman & Oldham 1975; Hackman & Oldham 1976). Therefore, an interaction effect between motivation and the corresponding governance structures and mechanisms can be expected. An interaction effect is thought “to exist when the effect of the independent variable on the dependent variable differs depending on the value of a third variable” (Jaccard & Turrisi 2003: 3). For example the existence of a strong copyleft license (independent variable) should influence the happiness of a developer with his project involvement (dependent variable) positively but only if his pro-social motivation is high (interaction effect). A developer motivated by reputation does probably not care about the license of a project. For such a developer the license should have no influence on how happy he is with project involvement.

The satisfaction with participation in FOSS development is not only influenced by the interaction of motivation and governance structures respectively mechanisms but also by these variables themselves. Such effects must be distinguished from the interaction effects. Additionally, in a large-scale study it was found that gender, marital status, age, education, religious beliefs, health, household income, and employment status influence volunteerism (Day & Devlin 1996: 44-47). Gender (see chapter 2.4), health, and religious beliefs are not thought to be important variables for explaining participation in FOSS development. However, other demographics variables such as age, education, household income, and employment status are thought to influence the decision to participate in FOSS development. This suggests that demographic variables are also important for the satisfaction with participation in FOSS development. Therefore, the interaction of motivation and governance structures respectively mechanisms should explain the satisfaction with participation in FOSS development after having controlled for demographics, motivation, and governance structures respectively mechanisms.

[543] This does not hold true for projects initiated by commercial firms. Projects started or dominated by commercial firms may follow a different pattern. However, such projects are not discussed in this study (see chapter 1.6).

Hypothesis 29: *After having controlled for demographics, motivation, and governance structures and mechanisms interaction effects between motivation and governance structures and mechanisms should explain satisfaction with participation in FOSS development.*

8.2.4 Hypotheses about motivation and the type of engagement

It is often assumed that in FOSS projects documentation, localization, and field support are poor (Fitzgerald 2005: 99; Weinstock 2005: 144). Alan Cox, a known FOSS hacker, responds to a question on what can be done to improve documentation (Linux Format 2005):

> *Well, you need to persuade more people that writing documentation is cool. We simply don't have people who enjoy doing that as part of the software community yet. There are clearly lots of people who enjoy writing documentation; documenting things accurately and precisely. Just look at Wikipedia: it's a huge community around documenting.*
>
> *It's getting better in some ways because obviously the big vendors are getting requests from their customers saying, "We've got this Linux thing, it's great, it's wonderful and it's faster than their old system, but when we used to run Solaris the manuals were brilliant." And so they're banging on the building saying, "Look, we love your product, but we want the manuals first. You know, the lack of documentation is costing us money and causing us problems; we don't like that." So there is a commercial pressure as well, which I think will help improve the documentation.*
>
> *… writing documentation appears to be something the community is bad at, but that's fine, because there are companies who like writing books about products and do a very good job of it.*

Based on the presented quotation one can conclude that writing documentation is at least now no fun for most developers. In fact, a survey about documentation for FOSS projects found that fun seeking is no major reason for contributing to documentation (Oram 2007). At the same time commercial firms write documentation because their customers request it. Based on this example it seems to be reasonable to assume that the type of engagement in FOSS development (e.g. writing documentation) is related to motivation (e.g. fun seeking). That different types of motivations are related to different tasks also supports heavily the assumption that different types of motivation in FOSS development are complementary.

Fun seeking

It was argued that per se all tasks are intrinsically motivating but that some tasks will be found more intrinsically than others (see chapter 3.2.1 and 5.6). It is argued here that fun seekers are primarily challenged by adding or developing features and by reviewing source code.

Hypothesis 30: *a) Fun seeking is positively related to adding or developing features.*

b) Fun seeking is positively related to reviewing source code.

Pro-social motivation

It can be assumed that pro-socially motivated FOSS developers engage in mundane tasks important for success but that are not very popular such as translating, creating graphics, writing how FAQ's, and donating. In fact, a survey found that the very most important reason to contribute to FOSS documentation is that contributors "care about the communities that they feel they are a part of" (Oram 2007).[544] Additionally, such individuals are thought to participate in tasks that are critical for spreading the FOSS Weltanschauung into a wider audience such as marketing and promoting.

Hypothesis 31: *a) Pro-social motivation is positively related to translating.*

b) Pro-social motivation is positively related to marketing and promoting.

c) Pro-social motivation is positively related to creating graphics.

d) Pro-social motivation is positively related to writing FAQ's.

e) Pro-social motivation is positively related to donating.

Peer recognition

Peer recognition is assumed to be related to tasks that include the interaction with peers such as reviewing/commenting source code and helping/answering questions of users.

Hypothesis 32: *a) Peer recognition is positively related to reviewing source code.*

b) Peer recognition is positively related to helping users.

Personal requirement

Personal requirement is thought to be related to tasks with an immediate practical benefit for the individuals. Therefore, this kind of motivation should be related to tasks such as reporting and fixing bugs as well as adding or developing features. For these tasks the relationship between effort and immediate benefit seems to be very favorable.

Hypothesis 33: *a) Personal requirement is positively related to reporting bugs.*

b) Personal requirement is positively related to fixing bugs.

c) Personal requirement is positively related to adding or developing features.

Reputation investment

It is argued here that not all roles in a project are similar informative about talent. Adding or developing features is considered more informative about programming talent than reporting bugs. Clearly visible to outsiders are tasks related to promoting and marketing.

Hypothesis 34: *a) Reputation investment is positively related to adding or developing features.*

---

[544] The second most important reason to contribute to FOSS documentation is investment in human capital (Oram 2007). Not surprisingly, investment in reputation is one of the least important reasons to contribute.

b) Reputation investment is positively related to promoting and marketing.

Investment in human capital

In chapter 3.2.2.3 it was discussed what skills are actually learned by contributing to FOSS development. Not surprisingly, basic and introductory programming skills and to look for and fix bugs are learned by contributing to FOSS development. Therefore, the tasks of adding or developing features and bug fixing should be related to investment in human capital. It was also discussed what is learned better in FOSS development than in formal computer courses. Astonishingly, an important aspect of learning in FOSS development is on the one hand to accept and respond to criticisms from others and on the other hand to evaluate the work of others and express personal opinions. One can therefore assume that the task to review and comment source code is also related to learning.

Hypothesis 35: *a) Investment in human capital is positively related to adding or developing features.*

b) Investment in human capital is positively related to bug fixing.

c) Investment in human capital is positively related to reviewing source code.

8.2.5 Hypotheses about motivation and community affiliation

In chapter 2.3.2 the differences between the free software and the open source community were discussed. It was argued that the main difference is in the Weltanschauung. The open source community considers access to source as a practical question while the free software community considers it as an ethical question. At the same time, both communities agree on practical recommendations and consider the same licenses as free respectively open. Therefore, only developers interested in the cause of FOSS respectively pro-socially motivated individuals should care to which community they belong. It is hypothesized that pro-socially motivated individuals feel to belong to the free software community and not to the open source community because the later is not based on ethical considerations.

Hypothesis 36: *a) Pro-social motivation is positively related to being part of the free software and negatively related to being part of the open source community.*

b) Other types of motivation are not related to community affiliation.

It was hypothesized (Hypothesis 12) before that pro-social motivation is positively related to participation in projects with a strong copyleft license. Consequently, being part of the free software community should also be positively related to projects with strong copyleft licenses.

Hypothesis 37: *Being part of the free software community is positively related to contribute to projects with a strong copyleft licenses.*

As already stated only the free software community is based on ethical considerations. The open source community is based on practical considerations. Based on this distinction it seems to be clear that subjects having a business relationship belong more likely to the open source community. Although the free software definition does clearly not exclude to earn money with free software[545] it is probably falsely interpreted in this way. Indeed, one reason to start the open source movement was “to market the free software concept to people who wore ties” (Perens 1999b: 173). In consequence, being part of the free software community should be negatively related to business interests.

Hypothesis 38: Being part of the free software community is negatively related to business interests.

8.2.6 Hypotheses about motivation and development phases

In chapter 6 it was shown that contingent on the development phase different motivations are served differently in the production of collective goods. Some individuals with a special motivation contribute at an early stage. Such individuals can hardly be motivated by the outcome of their action or by the circumstances. Other individuals jump on the bandwagon after they consider success likely. Similarly, Schweik and Semenov (2003) point out that the motivation to initiate a FOSS project is not similar to the motivation to join a project afterwards (see also chapter 2.8). Franck and Jungwirth (2003a: 14; 2003b: 418) argue similarly that the interplay between contributors with different motivations could follow certain patterns. They assume that FOSS projects cannot start without pro-social developers. Before reputation investors are attracted to contribute to FOSS projects it must have gained momentum. A similar pattern is outlined by organizational life cycles and group dynamics perspectives (see chapter 7).

In the following, it is assumed that there are four different development points (p^1, p^2, p^3, and p^4) that build three different phases of development. The initial phase occurs between p^1 and p^2, the intermediate phase occurs between p^2 and p^3, and the mature phase occurs between p^3 and p^4.

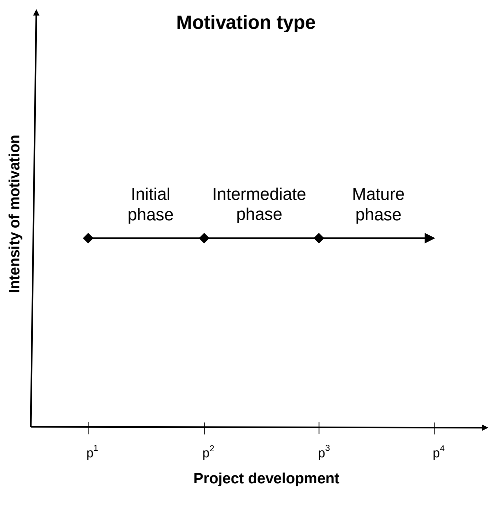


Figure 86: Motivation and development phases

[545] According to the free software definition it is even not possible to exclude commercial use. Yet the freedom granted by the GNU GPL does not refer to price. To understand this point the FSF suggests to “think of ‘free speech’, not ‘free beer’” (FSF 2006d).

Fun seeking

It is thought that autonomy decreases during software development (see chapter 8.2.7). At the very start of a project the developers do not have to care about users, backward compatibility, schedules, and the like. The more advanced the project gets the more restrictions impair autonomy. That is why it is assumed that the proportion of fun seekers decreases during development. Fun seekers contribute unconditionally because they enjoy programming and solving problems. They do not care whether success is likely or not or whether the software can be used safely on production machines. That is why fun seekers do not bother to start contributing to a project for which success is unpredictable. In consequence, the propensity of fun seekers to contribute to a project should decrease linearly $f(x) = -ax + b$.

Assumption: *The intensity of motivation related to fun seeking decreases constantly in project development. The intensity of motivation related to fun seeking in the different development phases is best described by the function* $-ax + b$.

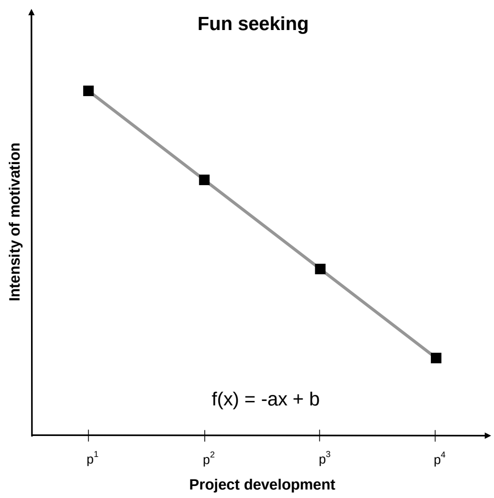


Figure 87: Fun seeking and development phases

Pro-social motivation

At June 15, 2006 122’520 projects were registered at the hosting platform SourceForge (2006a). Many of these projects will fail, development will peter out, and the software application will not contribute much to the cause of free software. However, pro-socially motivated developers want to see at least some probability for success. Such individuals are interested in the promotion of FOSS that is hardly achieved with unsuccessful projects. Therefore, these individuals include in their decision some sort of weighting of success probability. That is why pro-socially motivated developers will usually not contribute in the initial phase of a project.

Pro-socially motivated developers will stop contributing to a mature project if success is likely guaranteed by the contributions of others such as individuals motivated by personal requirements. Therefore, pro-socially motivated developers contribute prominently in the middle phase of FOSS projects. A second order function describes the presented pattern best.

Assumption: *The intensity of pro-social motivation increases in the initial phase, remains constant in the intermediate phase, and decreases in the mature phase. The intensity of pro-social motivation is best described by the function $f(x) = -ax^2 + bx + c$.*

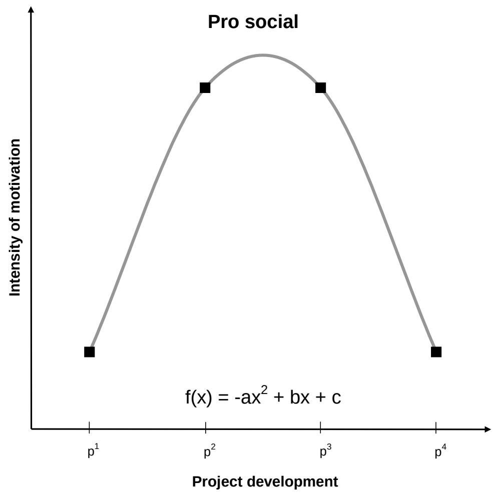


Figure 88: Pro-social motivation and development phases

Peer recognition

In chapter 7.2 group dynamics were discussed. It was suggested that groups follow a forming, storming, norming, and performing pattern. Based on this pattern of group development it is assumed that a sense of belonging to a group is high for juvenile and mature groups. In the storming phase and to lesser degree in the norming phase the feeling of being close to each other seems to be lower. Therefore, the motivation to contribute because of peer recognition should be high at the beginning of development and for very mature projects. Although it is argued that individuals motivated by peer recognition contribute in an early development phase they are not the very first. The very first lone developer of a project can hardly be motivated by peer recognition.

Assumption: *The intensity of motivation related to peer recognition decreases in the initial phase, remains constant in the intermediate phase, and increases in the mature phase. The intensity of motivation related to peer recognition is best described by the function $f(x) = ax^2 - bx + c$.*

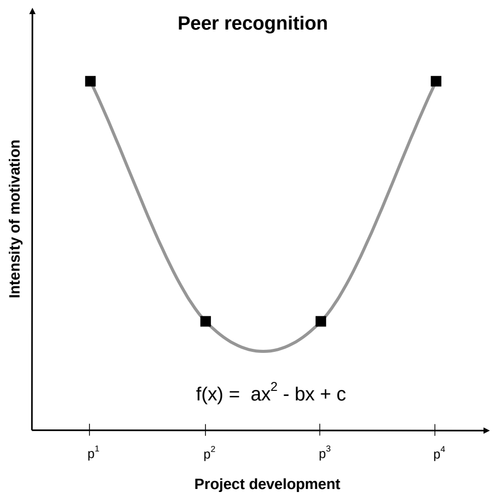


Figure 89: Peer recognition and development phases

Personal requirement

The relationship between personal requirement and development phases seems to be straightforward: The more advanced the development of a software application is the more valuable it is for a user. Therefore, the motivation to contribute to a FOSS project because of a personal requirement should increase constantly with project development.

Assumption: *The intensity of motivation related to personal requirement increases constantly during project development. The intensity of motivation related to personal requirement is best described by the function* $f(x) = ax + b$.

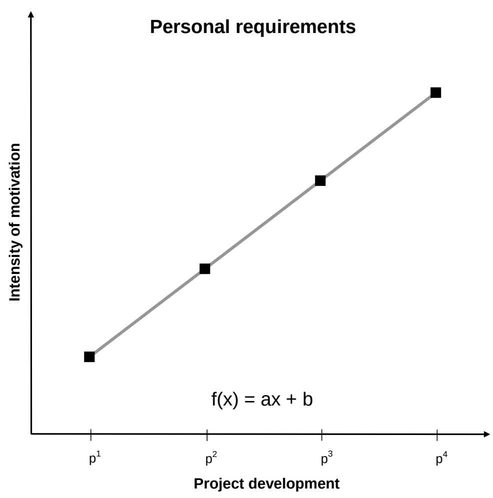


Figure 90: Personal requirements and development phases

Reputation investment

The value of a signal is related to its visibility and credibility. It is thought that visibility and credibility increase during development. Therefore, the motivation for reputation investors to contribute to a FOSS project should be positively related to the development phase of a project (see Figure 91). However, the value of a signal should be put in relation to the costs to produce it. The most valuable signals are the hardest to produce (see also Franck & Jungwirth 2003a: 14). Being the responsible maintainer for the standard Linux file system is a very valuable signal but enormously difficult to attain. With development going on it should be more and more difficult and therefore costly to get code merged. Producing a signal gets therefore increasingly costly. On the one hand competition for valuable signals rises and on the other hand the source code must be more and more proficient to replace existing source code. Therefore, the motivation of reputation investors to contribute to a FOSS project should flatten for mature projects. This is in line with the argumentation that in some cases motives must be weighted by the perceived likelihood that motivational goals are reached (see chapter 3.3 and chapter 5.4.2.1.1 on self-efficacy).[546] It was suggested that such a weighing is important for *investment in reputation,* for personal requirement, and for pro-social motivation.

Based on the arguments above it is assumed that the intensity of reputation investors to contribute to a project should follow a logarithmic function such as $f(x) = a * \ln(x) + b$.[547]

Assumption: *The intensity of motivation related to reputation investment increases in the initial phase, increases in the intermediate phase, and increases slightly in the mature phase. The intensity of motivation related to reputation investment in the different development phases is best described by the function* $a * \ln(x) + b$.

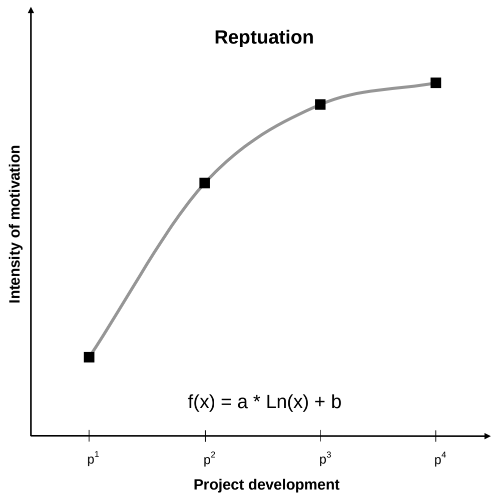


Figure 91: Reputation and development phases

Investment in human capital

In chapter 3.2.2.3 it was argued that developers learn by having users who have realistic demands, by peer review, by reading and analyzing source code, and by active participation instead of passive learning. Clearly, at the project start there are hardly users, a proper peer review, and source code to analyze. It is therefore

[546] However, see Eichenberger and Frey (1990: 270-274) who reviewed inconsistencies of observed choices with the predictions of expected utility theory. Such inconsistencies are for example visible in the Ellsberg paradox (Ellsberg 1961, 1963; Raiffa 1961; Roberts 1963) or in the Allais paradox (Allais 1953).

[547] That e is taken as a base and not another number has any other reason than that the natural logarithm is frequently used.

hard to learn something at the very beginning of a project. However, with development going on the potential to learn raises. In the initial phase developers get increasingly interesting feedback from users (see Figure 92). Yet, with development going on the basic functionalities are implemented and for users giving interesting feedback gets more and more difficult. Therefore, in the intermediate phase the possibility to learn (based on user feedback) decreases. At some point the source code matures and it gets technically advanced so that browsing it enhances learning. In the mature phase the possibilities to learn increase therefore again (based on browsing source code). In consequence, it is assumed that there are two phases of development in that the possibilities to learn are high: the initial and the mature phase. However, the reasoning for the two phases is not similar. The argumentation for the initial phase is based on feedback while the later is based on learning from browsing advanced source code.

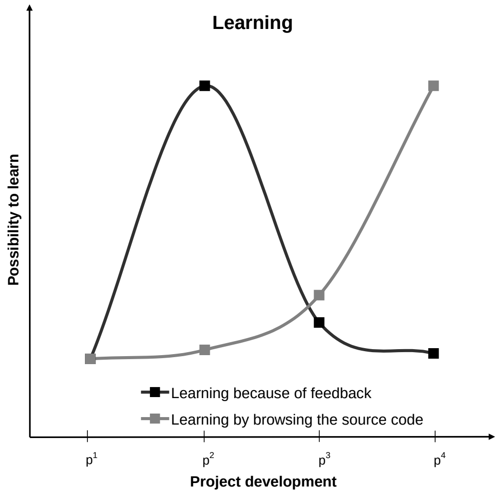


Figure 92: Learning and development phases

In consequence, the potential for learning increases after the project is started, decreases after the basic functionalities are implemented, and increases again after the source code is technically advanced.[548] This pattern is best described by a third order function.

Assumption: *The intensity of motivation related to human capital increases in the initial phase, decreases in the intermediate phase, and increases in the mature phase. The intensity of motivation related to human capital is best described by the function* $f(x) = ax^3 + bx^2 cx + d$.

[548] An important methodological remark about the above presented assumptions has to be made. The hypotheses concerning the relationship between motivation and development phases were stated in advance to the empirical analysis. Unfortunately, the hypotheses were made for a different number of development phases than actually used for the empirical analysis. There were too few subjects in the “planning phase” so that it had to be consolidated with the “proof of concept phase” (see chapter 11.2.2). For all motivation types except investment in human capital this adjustment in the number of development phases was compatible with the stated hypotheses. For example it was assumed that personal requirements increase linearly with development. The corresponding hypotheses can be adapted without any problems to a model with two, three, or four development phases. However, for human capital this was not possible and the corresponding assumptions had to be adjusted to the changed number of development phases. At this time the empirical results along general lines were known to the author of this study. This has most probably somehow influenced the above stated assumptions concerning human capital and development phases. In either way, the principle to state hypotheses in advance to empirical analysis has thereby been unintentionally violated. Strictly speaking, the collected data cannot be used anymore for hypothesis testing concerning human capital. Yet, in the following empirical analysis is also conducted for the assumptions related to human capital and development phases.

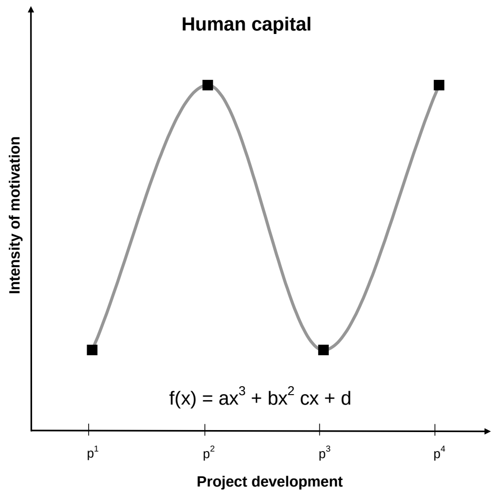


Figure 93: Human capital and development phases

Integrated hypotheses about motivation and development phases

In Figure 94 the assumptions concerning motivation and development phases are summarized. Based on these assumptions integrated hypotheses concerning motivational changes in different development phases are generated. These hypotheses are not geared towards specific types of motivation (e.g. reputation investment) but towards the idea of this dissertation at large: Are different types of motivation relevant in different development phases?

Predicted relationships between motivation and development phases				
		Development phases		
		p^1 → p^2	p^2 → p^3	p^3 → p^4
		Initial phase	Intermediate phase	Mature phase
Motivation	Fun seeking	Decreasing	Decreasing	Decreasing
	Pro social	Increasing	-	Decreasing
	Peer recognition	Decreasing	-	Increasing
	Personal requirements	Increasing	Increasing	Increasing
	Reputation	Increasing	Increasing	Increasing
	Human capital	Increasing	Decreasing	Increasing

Figure 94: Motivation and development phases

Hypothesis 39: *a) Motivation increases or decreases in different development phases contingent on the type of motivation.*

b) The distribution of motivation in the development phases can be described best by the specified functions.

8.2.7 Hypotheses about development phases and governance structures respectively mechanisms

One may argue that during the different development phases not only motivation changes but also project structures and mechanisms (see also Fombrun 1984: 211). In the course of development the difficulty for an outsider to get code merged is suggested to increase (see also chapter 8.2.6). On the one hand functions that are easy to implement are already included and on the other hand complexity of the project raises. The maturing of the source code is also positively related to the likelihood an application is included in a major Linux distribution. Mature applications also attract more developers and commercial firms. Additionally, the more mature a FOSS project is, the more likely it is missed by the community or the users if it were to disappear. It is also suggested that the number of different means to credit developers increases with project development.

Hypothesis 40: *During project development the fraction of projects included in a major Linux distribution increases.*

Hypothesis 41: *During project development it gets increasingly difficult for an outsider to get code merged.*

Hypothesis 42: *During project development the number of developers increases.*

Hypothesis 43: *During project development the involvement of commercial firms increases.*

Hypothesis 44: *During project development projects are increasingly missed by the entire community if they were to disappear.*

Hypothesis 45: *During project development the project is increasingly missed by the users if it were to disappear.*

Hypothesis 46: *The number of different means to credit increases with project development.*

It is thought that autonomy decreases during software development (see also chapter 8.2.6). At the very start of a project the developers do not have to care about users, backward compatibility, schedules, and the like. The more advanced the project gets the more restrictions impair autonomy. In consequence, it is also thought that the degree a project is missed by the developer if it were to disappear decreases with project development.

Hypothesis 47: *During project development autonomy decreases.*

Hypothesis 48: *During project development project are decreasingly missed by the developers if they were to disappear.*

In the planning and proof of concept phase of a project some structural characteristics are not yet built up. In this phase it is probable that no decision about the license has been made and that no credit mechanism has yet been established. With time going on it is likely that different means (e.g. credit file or website) to credit are established.

Hypothesis 49: *In the initial development phase the fraction of projects that have chosen a license increases.*

Hypothesis 50: *In the initial development phase the fraction of projects that have some sort of credit mechanism increases.*

It is argued that at the very beginning of a FOSS project there is not much feedback and not many decisions are explained (see also chapter 8.2.6). Communication between developers is hardly needed at this point of development because there is hardly any interference. However, with time going on developers get interested in each others work. This raises the need for feedback and explanations. After basic functionalities are implemented this need decreases. The most salient decisions are made. If the application is mature considerable effort is needed to improve it. This needs again an intensive exchange of information in form of feedback and explanations.

Hypothesis 51: *During project development feedback giving increases in the initial phase, decreases in the intermediate phase, and increases in the mature phase.*

Hypothesis 52: *During project development explaining decisions increases in the initial phase, decreases in the intermediate phase, and increases in the mature phase.*

8.3 Differences between threshold models and this model

The arguments and hypotheses in this dissertation are in the structure similar to the ones in threshold models. However, there are three main differences:

i. In threshold models of collective behavior individuals mostly rely on the number of other individuals who have already acted or contributed. The size of a movement is at the center of the reasoning (Yin 1998: 537). In the argumentation presented here the decision of individuals to contribute is based on the development phase respectively the production function of FOSS projects. The development phase and the production function determine the incentive structure for developers.

ii. Threshold models largely assume that the propensity of action increases with higher numbers of individuals. That more demonstrators join the opposition has the consequence that even more demonstrators join the opposition. The function describing participation in collective action is then monotonically increasing or even strictly monotonically increasing. In the model presented here this is not for all types of motivation the case. With development going on motivation may decrease.

iii. Elster assumes that the problem of collective action is solved in a two-step process. In the first step individuals motivated by altruism or pro-social behavior solve the start-up problem. In chapter 6.4.4 it was concluded that the start-up problem of the Monday demonstrations was solved by protestors who had a huge desire to live within the truth. In the terms of this dissertation these protestors were pro-socially motivated. In this study it is assumed that the start-up problem is solved by fun seekers. They do not care whether there is at least a minimal chance for success. It is assumed that pro-socially motivated developers do not want to waste their contributions to projects that perish. This difference in how the start-up problem is solved is most probably due to the characteristics of the public good. Although demonstrations and FOSS projects have some similarities they also have some distinctions. Again, one has to acknowledged that the specific characteristics of common goods matters.

Empirical testing and discussion of the hypotheses

Chapter nine sets the frame for hypotheses testing by discussing and defining the methodological and statistical assumptions. In ***chapter ten*** the survey respectively the method of collecting the data for hypotheses testing is presented. Afterwards the hypotheses are tested in ***chapter eleven***. Finally, based on the results of the study ***chapter twelve*** discusses lessons and further research questions.

9 Methodological and statistical assumptions: Setting the frame for hypotheses testing

Theoretical considerations of a study can be divided into two parts: the structural model that specifies the relationship between theoretical or structural constructs[549] and the measurement model that describes the relationship between constructs and items (Anderson & Gerbing 1982: 453; Edwards & Bagozzi 2000: 155). In the following chapter the later part is discussed. The basic assumptions and reflections about the theoretical constructs have been already discussed in chapter 8.

First of all, a general comment on methodology of FOSS research has to be made: It is not easy to conduct empirical research on FOSS development (Ghosh 2005: 24). Two characteristics of FOSS contribute heavily to this:

- *Firstly*, measurement is complex and ambiguous because FOSS development is often organized informally and without money as a mode of exchange. Related problems have already been observed in other areas. Spender (1996: 8) argues that „there is growing realization that the variables which are most theoretically interesting are those which are least identifiable and measurable“. He further states that “the dilemma for management is that, for the same reasons that competitors cannot replicate the firm’s knowledge, so the firm itself may not understand it well enough to exploit it effectively”. It seems to be reasonable to assume that scientists also struggle to understand the most interesting phenomena of FOSS development.
- *Secondly*, there are no objective sources such as the census on which indicators may rely on. There is for example no central database of FOSS project members that in turn inhibits proper sampling. This will influence the implementation of the survey of this study considerably (see also chapter 9.3).

9.1 Study design

Two important issues to be determined are the following ones:

i. Longitudinal vs. cross sectional research
ii. Observation vs. surveying

Below these issues will be discussed and afterwards determined.

[549] A construct is a conceptual term used to describe a phenomenon of theoretical interest (Edwards & Bagozzi 2000: 156-157). Although constructs refer to a real phenomenon they are no real in an objective sense. They are elements of a scientific discourse and “serve as verbal surrogates for phenomena of interest” (Edwards & Bagozzi 2000: 157). Latent constructs are not directly observable.

Longitudinal vs. cross sectional research

Longitudinal research involves a monitoring of the same items or/and subjects over a long period (see Figure 95, t^1, t^2 , t^3). For this study for each individual a motivational inventory could be produced. The decision to join or leave a project could be predicted conditional on the development phase. This would lead to very reliable data for a phenomenon that changes over time (see also chapter 9.7.3). One could also analyze the composition of the motivation of the developers in FOSS projects over time (e.g. for project C in t^1, t^2 , and t^3). However, longitudinal research is expensive, difficult to handle, and most notably time consuming (Bailey 1994: 36-37).

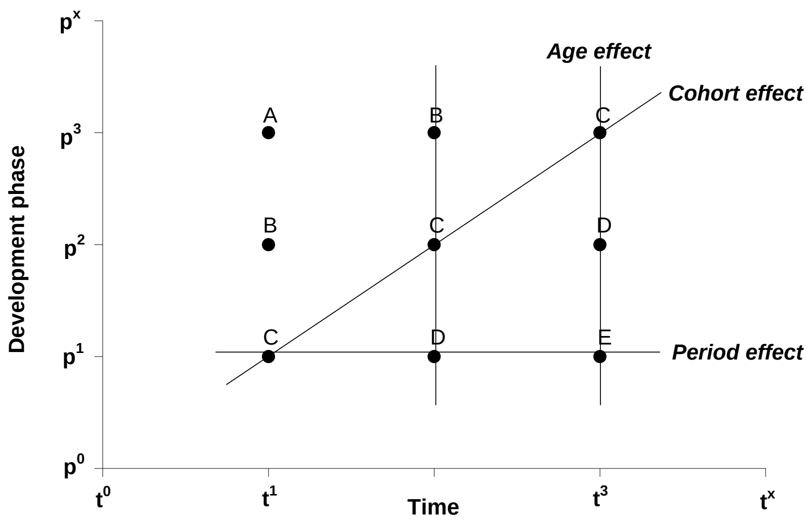


Figure 95: Period, age, and cohort effects

An alternative to longitudinal research are cross sectional studies. To study development phases respectively the "age" of projects (e.g. C, D, E; see Figure 95) a snapshot of projects in different development phases (p^1, p^2 , p^3) is taken. It is hypothesized that the motivation in projects C, D, and E are different due to the different development phases the projects are in. A problem related to such cross sectional research is that it is impossible to extract period effects and to distinguish between age and cohort effects (Chen, Wong, & Lee 2001; Glenn 1976; Gokhale, Kotlikoff, & Sabelhaus 1996; Hobcraft, Menken, & Preston 1982).[550] In context of this study this could lead to the problem that the findings (the different motivation in projects C, D, E) are perhaps due to a period effect at the time the survey was conducted and not to different development phases. It may be that a relationship between motivation and development phases were correctly observed at the time the survey was conducted (p^3) but that the very same relationship would not have been found before ($p^{1,2}$) or afterwards (p^x).

[550] According to Glenn (1976: 900) "age effects are produced by influences associated with the aging process, period effects by influences associated with each period of time and cohort effects by influences associated with membership in each birth cohort". A trivial example for an *age effect* may be the preference for sitting on a garden bench. While younger people do not like in general to stay calm older people do. Most probably this has always been true, even 100 or 1000 years ago. The dislike to throw away food is in some cases probably determined by a *cohort effect*. Individuals who have experienced a war or severe famine will not throw away food even if scarcity has long gone before. The admiration for a sportsman who won a competition is most likely a *period effect*. Individuals of all age admire the sportsman. However, this admiration will probably not last for the next decade.

It is also not possible to distinct between age and cohort effects. For example the Linux Kernel project (let us assume that project C is the Linux Kernel) was started in 1991[551] (=cohort) and is a mature (=age) project[552]. Its specific structure at t^3 may be due to an age effect or in the words of this study due to the development phase p^3. However, another possibility for explaining the results is that the found relationship is due to a cohort effect. In this case it was the "spirit" of the early nineties (t^1) or the emerging FOSS community that coined the structure of the Linux Kernel project. If so, the projects (e.g. project E) at t^3 in the p^1 phase will not have the same structure as now mature projects if they reach maturity (p^3) in t^5.

Mainly due to time constrictions of a dissertation it was not possible to conduct a longitudinal study. It was therefore decided to conduct a cross-sectional study. This decision was facilitated by the subjective presumption that FOSS development is not vitally influenced by period effects. The more serious problem seems to be to distinguish between age and cohort effects. Although the FOSS community does not have a long history[553] it may be that projects started at beginning of the FOSS movement are different from projects that started later. This possibility cannot be ruled out here. More corresponding research in the future is therefore needed.

Observation vs. surveying

The observation of actual behavior is one possible way for collecting data.[554] Marshall (1910: 15) argues:

> *It is essential to note that the economist does not claim to measure any affection of the mind in itself, or directly; but only indirectly through its effect. No one can compare and measure accurately against one another even his own mental states at different times: and no one can measure the mental states of another at all except indirectly and conjecturally by their effects. … If then we wish to compare even physical gratifications, we must do it not directly, but indirectly by the incentives that they afford to action.*

According to Marshall (1910: 15) it is for example neither possible to compare the pleasure two persons derive from smoking nor to compare the pleasures a person derives from smoking at different times. It is, however, possible to argue that a man in doubt whether to spend money for a packet of cigarettes or a cup of tea expects from them equal pleasures.

In the case of FOSS research observation of actual behavior prominently leads to documentary analysis that could include for example the analysis of emails, IRC communication, credit files, and CVS data (see e.g. Ghosh & Prakash 2000; Haefliger, von Krogh, & Spaeth 2007). This eliminates response biases that occurs if subjects answer questions in the way they think the researcher wants them rather than according to the truth. In surveys verbal reactions are researched based on the assumption that they reflect actual behavior and attitudes adequately (Likert 1932: 32). This is certainly not always the case. Response biases are especially problematic if there seems to be normative or morally correct answers for topics such as sex or suicide (Bailey 1994: 116). Another sensitive area besides social desirability is illegal behavior. Respondents may fear to tell

[551] Linus Torvalds (1991) announced at August 25, 1991 a free operating system. He stated that he was working on the operating system since April 1991.

[552] That Linux is mature is a subjective assumption of the author.

[553] Some notable FOSS projects started already in the early eighties. For example the GNU Project on which tools and libraries of many Linux distributions rely started in 1983 (Stallman 1983). However, most FOSS projects started more than a decade later. In either case these projects do not have a history that spans over generations of programmers.

[554] For a critique on motivational research based on self-reporting questionnaires see Bong (1996).

the truth because of possible prosecution. Yet another area in which response bias are important are questions related to behavior one is not self-conscious (Alexander & Becker 1978: 93). A related problem is that questions may be too difficult or abstract to answer (Finch 1987: 105).[555]

However, there are also some disadvantages of observation and advantages of collecting data by a survey. *Firstly*, there are problems of validity. It is not easy to find observational data that matches the construct to be researched exactly. Even simply counting lines of source code as a proxy for effort may be highly problematic: Different programming languages produce for the same function a different amount of lines of source code (Stevenson 1995). This problem may even exist within projects because it is possible to use different programming languages for different modules. Although there are ways to correct for such differences the problem cannot be solved completely. With programming tools a developer may also produce with a few mouse clicks automatically a big amount of source code for simple functions. Writing code is additionally not the only task a developer has to master. In FOSS development other tasks like for example testing are important as well. A developer who only writes source code but does not test it increases the amount of written source code. In consequence, measuring effort by counting lines of source code would be misleading. It is also possible that some part of the code is actually reused from another project and therefore not written by the author himself. Another problem is that lines of code may be correlated to effort but not to excellence. Skilled programmers may code the same functionality with less lines of code. In this case excellence could be negatively correlated to lines of code.

Secondly, if not manifest but latent constructs such as motivation are researched the problem arises that the observed behavior may have very different possible causes (see also Bailey 1994: 245). While it is perhaps possible to compute the amount of lines of code a developer contributed it is hard to figure out why the subject did so. Certainly, one may control for different diverging causes but this is practically not always possible. Measures of intrinsic motivation based on observations are by far not perfect. One measure for intrinsic motivation actually used in research is for example the time subjects spend voluntarily in an experiment on the target activity during a non-reward respectively free choice phase (Cameron, Banko, & Pierce 2001: 4). However, one may find several reasons why an individual does so without referring to intrinsic motivation. It is possible that extrinsic motivation manipulated in the experimental phase persists into the free-choice period. The need to abstract from a true value is therefore not restricted to surveys. The measurement of temperature in a room or the length of a line is also an abstraction of the true value (Thurstone 1928: 530). Therefore, too much faith in observational data should be avoided. Attitudes and behavior are not always related to each other in an obvious way (for examples see Lea, Tarpy, & Webley 1987: 8).

In a meta-analysis Deci et al. (1999: 646-656) concluded that the effect of rewards on intrinsic motivation was similar whether motivation was measured *by survey* or *by observation*. However, they argue that the best way assessing intrinsic motivation is to combine free-choice behavior (observation) with self-reported interest. Only if there is a correlation between the two measures the measures should be considered appropriate (Deci, Koestner, & Ryan 1999: 655). If it is not possible to use both measures Deci et al. prefer free choice measures respectively observation. In another meta-study on the effects of rewards on intrinsic motivation Cameron et

[555] A convenient solution to this problem is using real-life decision making respectively a vignette study that has been initially developed by Rossi (1974). The first idea to research the topic of this dissertation was to use such a vignette design. However, this idea was waived.

al. (2001: 22) found substantial differences between research that based on free-choice intrinsic motivation (observation) and research that based on self-reported intrinsic motivation.

To combine observational data (manifest behavior) and survey data (latent behavior) is another way of conducting an empirical investigation (see also Deci, Koestner, & Ryan 1999: 655). However, this leads to the severe problem of anonymity. If one combines observations and surveys subjects have to be identified somehow.

It can be concluded that theoretically observational data may be preferred over self-reported data. However, observational data is not without its pitfalls and self-reported data does also produce reliable results. In the case of this study the operationalization of the hypotheses in an observational study seems to be very difficult. Mainly the operationalization of the different motivation types is too demanding. In consequence, it was decided to rely on a self-reported survey. Social desirability does not seem to be very important for the items in this survey. However, problematic is that the decision to contribute to a FOSS project may have been made unconsciously. It is then probably difficult for subjects to answer items related to their motivation.

9.2 Elements of statistical inference

According to Cohen (1992: 156) there are four elements of statistical inference:

- sample size (N),
- significance criterion (α),
- population effect size (ES),
- and statistical power (1 - β).[556]

For statistical models any of these elements are a function of the remaining three. To determine for example the necessary sample size for a study one has to set significance, ES, and statistical power. Usually, the sample size is computed by taken significance and statistical power as given by convention and to set the effect size to the minimum that is theoretically or practically meaningful.

The approach to only look at the significance level is highly dominant although there has been in some scientific areas such as medical research a substantial shift to include power (Singh 2006). There are several disadvantages of only relying on the significance level. One disadvantage is that researchers often think that a P value of 0.001 reflects a larger effect or is more important than a P value of 0.05 (Singh 2006: 149). It is often also thought that equal P values of two studies with a large and a small sample size reflect the same effect size. In the following, all elements of statistical inference will be discussed in turn. Afterwards, corresponding decisions for this study are made.

[556] There are three major approaches to hypothesis testing and inductive inference: the Bayes, the Fisher, and the Neyman-Pearson approach (Sedlmeier & Gigerenzer 1989: 315). Statistical power is only relevant in the Neyman-Pearson approach. In practical usage inferential statistics is a hybrid model of these different approaches (Sedlmeier & Gigerenzer 1989: 314). The statistical procedures in this study follow the common hybrid model.

Sample size

Estimating an adequate sample size is always an important problem in designing research (Bailey 1994: 82). As discussed above the optimal sample size is a function of effect, power, and significance. Additionally, sample size depends on the statistical test (Cohen 1992: 158). Contrary to common belief the sample size is practically independent of the population size (Deming 1960: 29 and 387).[557] Optimal sample sizes to research food preferences in the United States and Switzerland are ceteris paribus basically the same although the population of the United States is approximately 301,1 millions and the one of Switzerland only 7,6 millions (July 2007 estimates CIA 2007d).

An important question in research design is whether the purpose of a study is to estimate the true parameter values precisely or whether the intent is to detect and map out the main relationships (Baroudi & Orlikowski 1989: 98). If you increase the sample size the confidence interval of the true effect decreases and the result gets more precisely. In consequence, the more precisely a result should be the higher sample size is needed.

It is not always possible to draw optimal sample sizes. There may be for example a shortage of subjects or objects that can be researched (March, Sproull, & Tamuz 1991). Although a power company may have a hypothesis how to avoid nuclear accidents there are (luckily) not enough catastrophes for conducting a statistical test based on a frequency distribution.[558] It may also be that subjects are hardly accessible such as CEO's of multinational corporations (Baroudi & Orlikowski 1989: 102). In practice costs are often a factor that limits the size of a sample (Deming 1960: 10). Large sample sizes are expensive in terms of time, effort, and money (Baroudi & Orlikowski 1989: 98). In some areas such as car crash tests a large sample would destroy the product that was supposed to be sold. There may be also ethical considerations to keep the sample size small to avoid possible harm. Corresponding examples are found in psychology and drug clinical trials.[559]

Significance criterion

The significance criterion characterizes the probability of rejecting the null hypothesis[560] wrongly and committing a Type I error (Neyman & Pearson 1967: 190).[561] This error is a false negative or in other words an

---

[557] Exceptions are very small populations. However, there are different definitions what small populations are. In a conservative approach Bortz (1993: 90) suggests to correct sample sizes if the population divided by the sample size is smaller or equal 100. At the FOSS hosting platform SourceForge 1'300'000 developers are registered (SourceForge 2006a). If one assumes that only 10% of these developers are active, that there are no FOSS developers not registered at SourceForge, and that the sample size is 1000 no correction is needed. Even if the real size of the population of FOSS developers is not known it seems to be clearly justifiable to assume that the population is so large that no correction is needed.

[558] In such cases risk estimates have to rely on propensity (Gigerenzer 2002: 26-27). In the words of Frank Knight (1921 part 3, chapter 4, 36) a corresponding probability judgment cannot rely on statistical calculation but on a priori calculation. An example of a priori calculation is the NASA estimation of the probability of a failure of the Space Shuttle with a loss of vehicle and of human life. The corresponding estimate of such a loss was thought to be 1 in 100,000 flights (see Footnote 23). The same probability estimated by statistical calculation is about 1 in 57 flights.

[559] Clinical trials have usually four phases in which the number of involved persons are constantly increased (U.S. National Institutes of Health 2005). In phase I a small group of 20-80 subjects is researched to evaluate safety (for a tragic event in this phase see Suntharalingam et al. 2006). In phase II the sample size is increased to 100-300 subjects to research effectiveness and again safety. In phase III the drug or treatment is given to a large group of 1000-3000 subjects. In phase IV the drug or treatment is released to the public.

[560] Correctly, H**o** is <u>not</u> the ***null*** hypothesis but the ***origin*** of an event (Neyman & Pearson 1967: 186). However, by usage researcher got used to this naming. Nevertheless, one should not assume that null hypothesis states always that there is null difference. It may be that the null hypothesis states that the mean of the population A is different from the one of

error of rejecting something that should have not. Associated with the significance level is the P-value. While the significance level is decided before the data is collected the P-value is observed in the data. The P-value is the probability that given the null hypothesis is true the findings are the result of pure chance (Gibbons 1985: 10-15). If the P-value is below or if it is equal to the significance level the result is significant and the null hypothesis can be rejected for now.

Normally, it is taken for granted that the significance should be 0.05 (see Fisher 1971: 13) or lower. Fisher (1925: 46) is credited to use the 0.05 level first (Sauley & Bedeian 1989: 336-337). This significance level means practically that in 5 out of 100 studies with no true effect the null hypothesis is rejected although it should not have been. Baroudi and Orlikowski (1989: 102) conclude that "researcher have begun to question the utility of such an approach and have indicated a need to rationalize research by viewing alpha as variable … subject to adjustment". Fisher (1925: 504) explicitly stated that the significance level of 0.05 is his personal preference. Therefore it is not surprising that the significance level of 0.05 is called arbitrary (Cohen 1990: 1307; Cohen 1992: 156), a matter of personal taste (Yule & Kendall 1950: 472), and not based on logic (Winer 1962: 13).

There are no right or wrong significance levels (Ferguson & Ketchen 1999: 387) and one "is free to choose whatever level seems appropriate" (Camilleri 1962: 176). To be appropriate the significance level should be chosen based on research design (Sauley & Bedeian 1989) and by assessing the effects if the related error is made (Baroudi & Orlikowski 1989: 102). Rejecting the null hypothesis that a drug has an adverse effect threatening life is a different thing than rejecting the null hypothesis of a newspaper that Switzerland gets two gold medals next Olympic Games. Sauley and Bedeian (1989: 339-342) suggest considering the following parameters if an appropriate significance level is selected:

- ***Sample size:*** If the sample size is large enough any relationship will be found significant and vice versa. Therefore, the significance level for large samples should be lower to exclude that everything gets significant.
- ***Effect size:*** The higher an effect is the easier it gets significant. To adjust for this phenomenon a researcher should set a low level of significance if there is a high effect.
- ***Measurement error:*** If there is a high degree of measurement error (coding error, interviewer fatigue, or memory lapses) the significance level should be lower to avoid interpreting random variation as significant.
- ***Consequences:*** The more severe the consequences of a type I error are the lower significance level must be chosen.

---

population B. The corresponding alternative hypothesis is that there is null difference between the two population means.

[561] The errors of the first and second kind are frequently discussed. Mosteller (1948: 61) recognizes additionally an error of the third kind namely "correctly rejecting the null hypothesis for the wrong reason". Kimball (1957: 134) also discussed "the error of the third kind" which "is the error committed by giving the right answer to the wrong problem". Similarly, Mitroff and Featheringham (1974: 383) define the third error "as the probability of having solved the wrong problem when one should have solved the right problem". Raiffa (1968: 264) proposed that the error of the fourth kind is "solving the right problem to late" while Marascuilo and Levin (1970: 398) propose that the error of the fourth kind is "defined as the incorrect interpretation of a correctly rejected hypothesis". The later authors compare this error "to a physician's correct diagnosis of an ailment followed by the prescription of a wrong medicine".

- ***Coherence of underlying theory:*** If a hypothesis runs against theory, common sense, and other empirical research the significance level should be lower to avoid further research in a fruitless area to save money, time, and effort. The same argumentation applies in the case in which the plausibility of a hypothesis is unknown. In consequence, a higher significance level is appropriate if the knowledge of a theory is neither established nor very new.
- ***Experimental control:*** With a higher degree of experimental control the error rate and the likelihood of alternative explanations decreases. This in turn allows choosing a higher significance level.
- ***Robustness of statistical test:*** If assumptions underlying a statistical test are violated a lower level of significance should be chosen (see also chapter 9.8). If the opposite is true or a test is robust against such violations a higher significance level is appropriate.

For Ferguson and Ketchen (1999: 387) the appropriate significance level is contingent on the state of theory development. They state in contrast to Sauley and Bedeian (1989: 341) that a relatively high significance level (e.g. $\alpha = 0.10$) is appropriate if a theory about a phenomenon is not developed while the significance level should be more conservative for a hypothesis that challenges an established body of knowledge. This is in line with the observation that it is very hard to achieve highly significant results for an unknown phenomenon. The less fine-graded the knowledge of a researcher is, the more difficult it gets to design the research. This argumentation of Ferguson and Ketchen seems to be more conclusive than the corresponding argumentation of Sauley and Bedeian.

In scientific research not only the adherence to the 0.05 significance level but also the significance criterion itself seems to be overvalued (Sedlmeier & Gigerenzer 1989: 315). William Sealey Gosset, the inventor of the heavily used t-test, writes in a letter to Egon Sharpe Pearson (1939: 247) that a significant result “seems to … be nearly valueless in itself”. The significance of a result does not tell the researcher anything about its importance. Olejnik and Algina (2000: 241-242) note that

> *… statistical significance does not imply meaningfulness. ‘Significance’ based on a statistical test provides information on the likelihood of finding the observed relationship by chance alone (sampling error). While statistical “significance” helps to protect the researcher from interpreting an apparently large observed difference as meaning a true difference between populations when sample sizes are small, it does not protect the researcher from interpreting a trivially small observed difference as meaningful when sample sizes are large. Small differences can be statistically ‘significant’ simply because of a large sample size.*

To control for “significance” it is also possible to use effect sizes that are more robust measures of theories’ effectiveness (Baroudi & Orlikowski 1989: 90).

Power

Statistical power (1- β) is especially important in replication studies (Tversky & Kahneman 1971: 107). It refers to the probability that a wrong null hypothesis is rejected as it should. The higher power is the less likely one makes an error of type II (β) which is a false positive. A type II error refers to an incident in which the null hypothesis should have been rejected but is not. Dyba et al. (2006: 745) observe:

> *Statistical power is an inherent part of empirical studies that employ significance testing and is essential for the planning of studies, for the interpretation of study results, and for the validity of study conclusions.*

Usually, statistical power is determined to be 0.80 (Cohen 1992: 156). However, the 0.80 level is yet another convention that should be based on the associated risks. According to Neyman and Pearson (1967: 190-191) the consequences of falsely accepting the null hypothesis (error type II) depends on the nature of the alternative hypothesis. Generally, the more it differs from the null hypothesis the more serious are ceteris paribus the consequences. Power values above 0.9 require large sample sizes that exceed available resources mostly (Cohen 1988: 55). Cohen (1988: 56) suggests setting β four times the significance level. This arbitrary convention is based on the assumption that making an error type I is four times more serious than making an error type II. Following this rule of thumb in a study with a significance level of 0.05 a β of 0.20 and therefore a power level of 0.80 should be chosen. However, one should note that Cohen (1988: 56) offered this convention "with the hope that it will be ignored whenever an investigator can find a basis in his substantive concerns in his specific research investigation to choose a value ad hoc".

Meta studies in psychology (Sedlmeier & Gigerenzer 1989), software engineering (Dyba, Kampenes, & Sjoberg 2006), research on performance of different organizational configurations (Ferguson & Ketchen 1999), and management research (Mazen et al. 1987) found that power is hardly discussed and generally very low. The consequences are impressive. Sedlmeier and Gigerenzer (1989: 312-313) conclude that the median power of the surveyed psychological studies is 0.25: "This means that the experimental conditions, such as number of observations, were set up in such a way that given a true medium effect, the research (null) hypothesis would nevertheless be 'confirmed' in 75% of the cases". The same low power level (below 0.3) is found in studies on software engineering (Dyba, Kampenes, & Sjoberg 2006: 750). For management research power is much more favorable (Mazen et al. 1987: 373). Studies deficient in statistical power are wasteful as well as pernicious (Tversky & Kahneman 1971: 107).

The reason for the low power found in the meta-studies presented above may be that researchers overestimate the power of their research constantly. Tversky and Kahneman (1971) showed in an experiment with members of the Mathematical Psychology Group and the American Psychological Association that researchers could not assess how sample size, power, significance level, and effect are connected to each other.

Effect size

With a sample large enough every relationship gets significant even if the effect is unimportant or minuscule. In consequence, significance should not be confused with importance. Effect size determines how large an effect must be for being of practical or theoretical importance. A diet that results in the loss of 1 gram is probably of no value for the individual even if this effect is highly significant. To be relevant the diet should have, let us say, a minimal effect of 1 kilogram. Beside its high importance for meta-studies (see for example Hedges & Olkin 1985) effect size is therefore very relevant for practical purposes. It is practically not sufficient to know that a new organizational process is significantly better than the old one. Switching to the new process is accompanied with costs that should only be beard if the effect of the performance improvement is above a certain minimum.

The larger ceteris paribus an effect is the more likely a test will detect it (Ferguson & Ketchen 1999: 387, see also).[562] Thereby a relationship to sample size is identified. Larger effects need ceteris paribus lower sample sizes. It is easier to detect that a diet works if it results in an average loss of weight of 10 kilogram than if it

[562] If the null hypothesis is true the effect size is zero.

results in an average loss of weight of 1 gram. An average loss of weight of 1 gram may be likely due to chance while the same thing unlikely happens if the average loss is 10 kilogram. Figuratively, at a safari it needs a binocular with lower amplification for spotting an elephant than spotting an elephant shrew if the distance remains constant.[563]

Although effect size is probably more important than the other parameters it is the most difficult to estimate (Baroudi & Orlikowski 1989: 90). To specify effect size one has to be consciousness about the magnitude of the phenomenon (Cohen 1992: 156). To do so effect size can be estimated based on the proportion of explained variance in prior research (Baroudi & Orlikowski 1989: 90). Another possibility is to conduct a pilot study to estimate the effect size. If both alternatives are not possible one can use proxy effect sizes (Baroudi & Orlikowski 1989: 91). Cohen specified the three effect size proxies small (0.1), medium (0.25), and large (0.4) which are comparable for different statistical tests.[564] Medium effects represent effects that are "visible to the naked eye of a careful observer" (Cohen 1992: 156). Small effects are specified above being trivial. Large effects are in general so apparent that the use of statistical tests is virtually superfluous (Cohen 1965: 97, cited in Baroudi and Orlikowski 1989: 90). Cohen (1988: 12) makes clear that the presented qualitative definitions of large, medium, and small effects are arbitrary and therefore a source of danger.

Another approach to specify the effect size is to set it to the minimal relevant size. If one knows that in practice only a big effect matters a large effect can be presumed. With such a research design one does most probably not discover a small effect. However, this is no problem because such an effect is thought to be irrelevant anyway.

Cohen (1988: 13) supposes that small effects are the norm for research about personality, social, and clinical-psychological research. He explains his assumption by "the attenuation in validity of the measures employed and the subtlety of the issues". Cohen supposes further that because of the good experimental control and potent variables research in the field of sociology, economics, and experimental and physiological psychology are characterized by large effects. There is some evidence that management research is no exception concerning effect size (Mazen et al. 1987: 376).

How power is related to the significance criterion, to effect size, and to sample size: Graphical representation

To exemplify the relationship between the discussed elements the influence of the significance criterion, the effect size, and the sample size on power is graphically presented in Figure 96 until Figure 99. In this example it is hypothesized that two groups of a population differ in a certain characteristic. The null hypothesis states that there is no difference between the two populations. In each Figure the standard deviations from the mean of the characteristic in the two groups are presented.

- Comparing Figure 96 and Figure 97 one can see that if one lowers the significance level the power gets lower as well.
- Comparing Figure 96 and Figure 98 one can see that if the effect gets smaller the power gets smaller as well.

---

[563] Elephant shrews (elephantulus spp.) are little insectivorous mammals also native to the Serengeti (Tanzanian Parks Authority).

[564] For a short overview over different interpretations of effect size see Kotrlik and Williams (2003: 5).

- Comparing Figure 96 and Figure 99 one can see that with a higher sample size the standard deviation is reduced which leads to higher power.

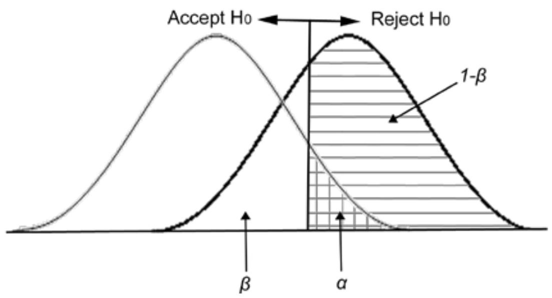


Figure 96: Power (1- β), type I (α) error, and type II (β) error for a directional hypothesis.

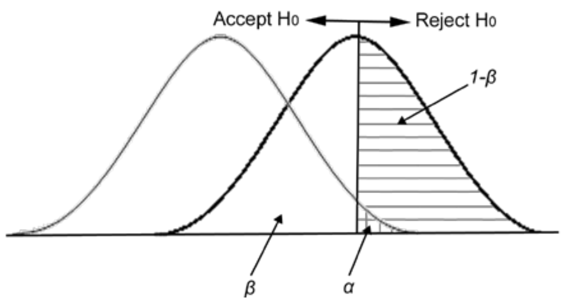


Figure 97: Lower significance level and power

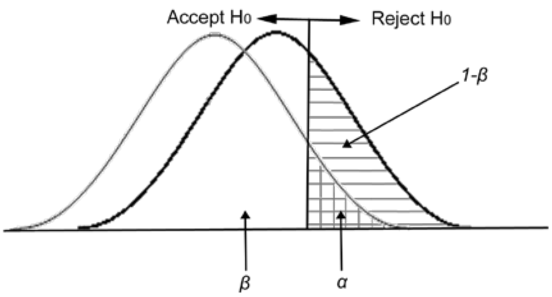


Figure 98: Smaller effect size and power

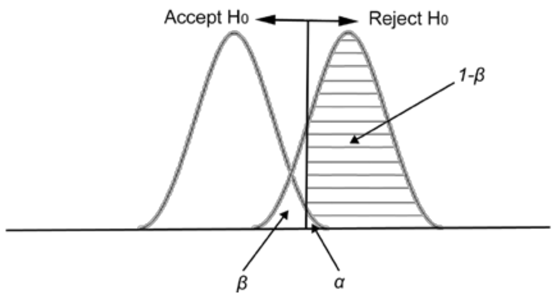


Figure 99: Higher sample size and power

Significance criterion, effect size, statistical power, and sample size in this study

For planning a study the researcher has to set the significance criterion, estimate the effect size, set statistical power, and solve for the necessary sample size (Baroudi & Orlikowski 1989: 89). This is done below.

Significance criterion

In this study the consequences of an error of the first kind are not especially grave so there is no need to select a very low alpha. Additionally, that motivation is related to the development phase of FOSS projects is a new hypothesis. The hypotheses therefore do not run against theory or other empirical research because there is none. It is also not thought that that the hypotheses run against common sense but this may be a subjective opinion of the researcher. However, it may be possible that the measurement error is above the average.

Because of the argumentation above it seems to be reasonable not choosing a conservative significance level. In the following, the significance criterion is set as 0.10. This implies, however, that there will be a 10% chance that a significant result is not based on a true effect but on chance.

Effect size

There is no previous research available to estimate the magnitude of the phenomenon. Due to restrictions of scare resources a pilot study is not feasible. The only way to set the effect size is to take proxies and rely on minimal effect relevance. For this preliminary stage of research it is not likely to find a large or even a medium effect. I will be difficult for a non-mature theory as the one presented here to be so specific and accurate that every fine nuance is enclosed. Therefore, only rough hypotheses comprising some noise can be stated. Additionally, it must be assumed that the validity of the measures (e.g. motivation) is low. Therefore, a small effect must be expected. However, to find small effects a huge sample size is needed that overloads financial and time resources of this doctoral thesis. Therefore, the research design is adjusted to find a medium effect.

As argued above it is more convenient to search for small effects. If in reality the effect of the observed phenomenon will be lower than a medium effect the true power of the study will decrease if the significance level is not relaxed. In such a case even if there is a true phenomenon the null-hypothesis cannot be discarded.

Power

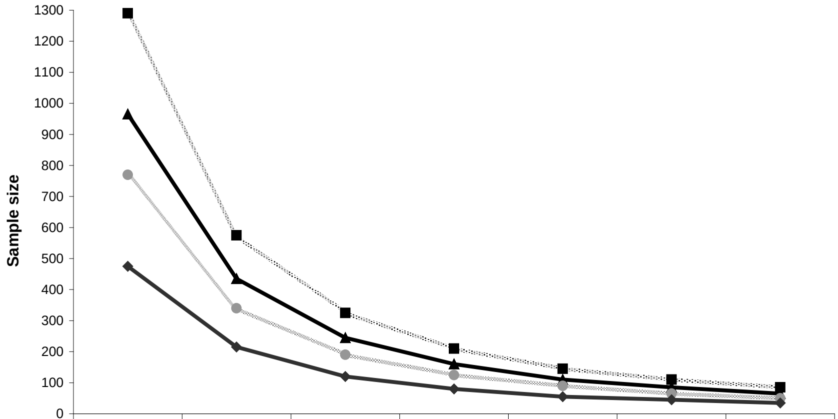
As a rule of thumb Cohen (1988: 56) proposes setting β four times higher than the significance level. This induces a 0.6 power level. For this preliminary study making a type II error should be avoided why a higher power level is chosen than suggested by Cohen. A probability of 20% making a type II error respectively a power level of 0.8 seems to be acceptable.

Sample size

In Figure 100 optimal sample sizes for five group one-way ANOVA[565] tests contingent on effect, power, and significance level are presented. The ANOVA test was chosen because it will be used to test some aspects of the main hypotheses that are concerned with the change of motivation in five development phases of FOSS projects.[566] Based on the decisions concerning the significance level (0.1), effect size (medium), and power (0.8) the theoretical optimal sample size is 160 (see Figure 100). Practically, the sample size should be bigger because of missing values and because an equal distribution of subjects into groups can hardly be assumed. Therefore, it is thought that a target sample size of 200 subjects should be appropriate. As a comparison the optimal sample size for the significance level 0.05 is presented in Figure 101 as well.

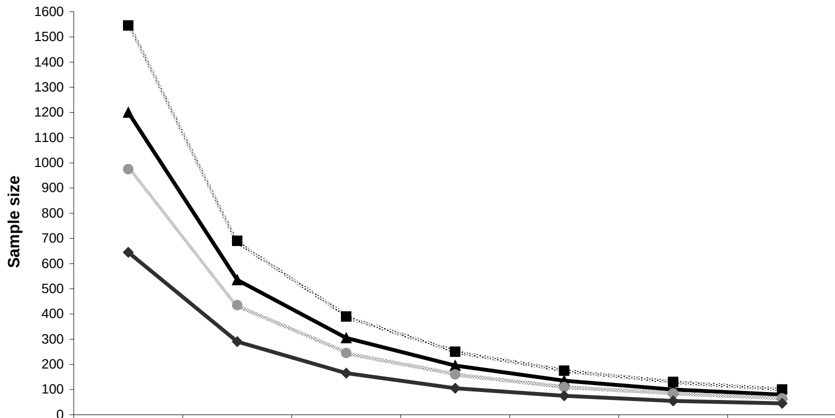
[565] Analysis of variance (ANOVA) is a procedure to investigate the relations (significance and magnitude) of a dependent variable to each of the independent variables (Spector 2001: 20). An ANOVA is similar to multiple regressions but it is easier to set up and interpret. For the first use of variance in statistics see Fisher (1918).

[566] In fact, because there were too few subjects in the "planning" phase it has been collapsed with the "proof of concept" phase (see chapter 11.2.2). Therefore, statistical tests are only conducted for four groups. The optimal sample size for a four group ANOVA (significance level =0.1, effect size = 0.25, and power = 0.8) is 144 and therefore smaller than computed for five groups. In consequence, the optimal sample size has been overestimated which is no methodological problem.



	Small effect 0.1	0.15	0.2	Medium effect 0.25	0.3	0.35	Large effect 0.4
Power 0.9	1290	575	325	210	145	110	85
Power 0.8	965	435	245	**160**	110	85	65
Power 0.7	770	340	190	125	90	65	50
Power 0.5	475	215	120	80	55	45	35

Figure 100: Optimal N for 5 group ANOVA as a function of significance (α = 0.1), power, and effect[567]



	Small effect 0.1	0.15	0.2	Medium effect 0.25	0.3	0.35	Large effect 0.4
Power 0.9	1545	690	390	250	175	130	100
Power 0.8	1200	535	305	195	135	100	80
Power 0.7	975	435	245	160	110	85	65
Power 0.5	645	290	165	105	75	55	45

Figure 101: Optimal N for 5 group ANOVA as a function of significance (α = 0.05), power, and effect[568]

Optimal sample sizes depend not only on power, effect size, and significance but also on the statistical test. If in a study different hypothesis with different statistical tests are researched there are different optimal sample sizes. However, sample sizes based on a five group one-way ANOVA are sufficient for other tests such as chi-square and correlation[569] if effect size, power, and significance are equal. A chi-square test with four degrees of freedom based on the same assumptions requires a sample size of 108 subjects (Cohen 1988: 264). For an analogous two-sided product-moment correlation 68 subjects are needed (Cohen 1988: 101).

[567] Cohen (1988: 387). The sample sizes are only optimal if N is for every group equal.

[568] Cohen (1988: 384). The sample sizes are only optimal if N is for every group equal.

[569] For a discussion of correlation methods see Cohen (2003) and Pedhazur (1997).

There is an unresolved discussion about minimal sample sizes for chi-square tests (Delucchi 1983: 166-168). While some authors recommend a minimal cell frequency of 1 other authors suggest a minimal cell frequency of 5, 10, or 20 (Delucchi 1983: 167). However, it is usually thought that no cell in the table of a chi-square test should have an expected frequency below 5 (Cochran 1954: 418; Yates 1934: 217). Cochran (1954: 420) suggest that there may be cells with an expectation below 5 if they count for not more than one cell out five. Yates (1934) suggests applying what is now known as the Yates correction for continuity if cell frequency is below five. For this study cell frequencies below 5 are attempted to avoid. For at least cardinal data this may be done by pooling categories.

Tradeoffs surfaced and debated

Whatever effect size, significances criterion, statistical power, and sample size is chosen there are always tradeoffs (Baroudi & Orlikowski 1989: 91-92). These tradeoffs should be consciously surfaced and debated. For this study this is done below.

The chosen power (0.8) and the significance level (0.1) seem to be appropriate. Problematic is clearly the chosen effect size. For a preliminary study it will be difficult to detect a medium effect (0.25). A short discussion what happens if the effect is lower than assumed is therefore necessary.

If the true effect is smaller than medium the power drops from 0.81 to 0.24 if the effect is small (see Figure 102). If the effect is slightly above 0.17 power is approximately 0.5. In this case the chance that a false null-hypothesis is not discarded is fifty-fifty.

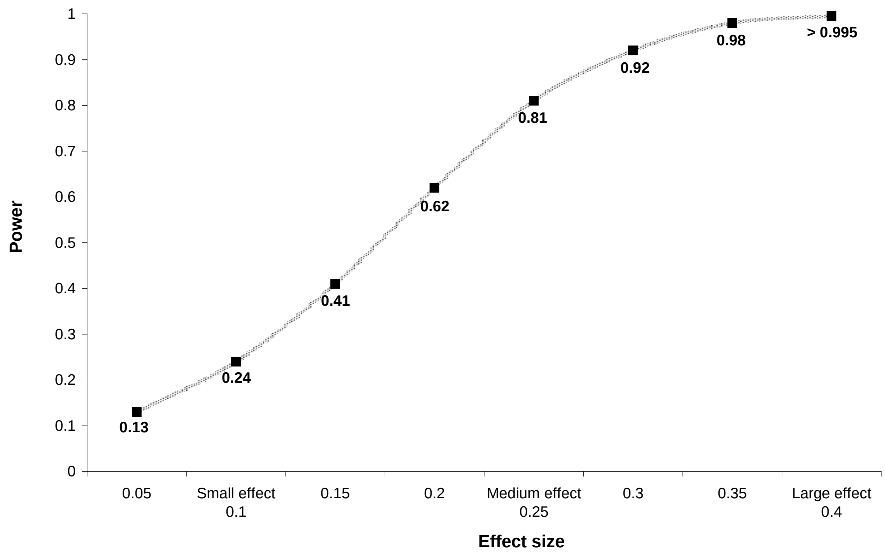


Figure 102: Power for a 5 group ANOVA as a function of significance ($\alpha = 0.1$), N (160), and effect[570]

If the effect sizes are lower than assumed some null-hypothesis will not be discarded even if the alternative hypothesis is true. Because the purpose of this study is not to estimate true parameter values precisely but to detect and map out the main relationships (Baroudi & Orlikowski 1989: 98) this is very regrettable but not dramatic. If the main null-hypotheses cannot be discarded but the results give an impression whether the

[570] Cohen (1988: 339).

general idea has some appeal or not at least a minimal goal is reached. In this case the pattern of results (Sauley & Bedeian 1989: 340) should show that the phase of development has an influence on developers motivation to join a FOSS project.

9.3 Sampling

There are two main classes of sample methods: probability and non-probability samples (Bailey 1994: 89). In the former case the probability of being selected to participate in the survey is known for each subject. This allows calculating sampling errors (Deming 1957: 10). In non-probability, also called judgment samples, the probability of selection is not known and the sampling errors must be settled by judgment.

Probability samples

The most known probability sample is the random sample. If this sample method is applied every subject has an equal probability of being chosen for the sample (Bailey 1994: 89). Random sampling has the huge advantage of canceling out some of the major biases and providing a statistical mean for estimating sampling errors (Bailey 1994: 90). However, to draw a random sample is often not possible (Baroudi & Orlikowski 1989: 99). To do so a reliable and exhaustive list of subjects is needed. From this list the researcher draws randomly subjects until the targeted sample size is attained. To do so three steps have to be executed:

i. Definition of the problem and the designation of the population
ii. Data file compilation of the whole population
iii. Selection of a sample out of the population

To define what qualifies a FOSS developer is not easy. The biggest problem is to find a definition that is theoretically sound and that is at the same time practically workable (Deming 1957: 31 and 83-84). It is not clear which contribution qualifies an individual as a FOSS developer. One possible solution is to consider that everybody is a developer if the individual thinks himself that he is a developer. However, this is certainly not a workable definition for a probability sample. The definition problem is increased by the fact that there is seldom a formal membership in FOSS projects.

Even more difficult than providing a sound definition seems to be the compilation of a reliable and unbiased data file of the population. As discussed in chapter 4 there is no central entity that organizes FOSS programmers. It is in fact a characteristic of FOSS communities that everybody may contribute. This allows a user to contribute only one small bug fix in his lifetime. It is hardly possible to find all FOSS projects or all contributors. There are over 120’000 projects hosted only on the FOSS hosting platform SourceForge.[571] The huge number of projects makes it therefore prohibitive costly to list all contributors. It is also impossible to find all the small FOSS projects not hosted on one of the known hosting platforms. Additionally, such a procedure does not reveal real identities but only email addresses. As a result there would be many invalid email addresses that could distort the sample. Some subjects may have several email addresses and identities. In consequence, these subjects have a greater chance being included in the survey than other subjects. Next to these minor methodological problems an attempt for listing subjects would be tremendously time consuming

[571] See footnote 93.

and expensive. For these theoretical, economical, and practical reasons it is hardly possible to compile a list of all FOSS developers. It is therefore practically not possible to draw a fully random sample.

The reasoning why a fully random sample is not possible holds true for all other probabilistic sampling methods like systematic or stratified random sampling. Additionally, for stratified random sampling there are two other problems. *Firstly*, it would be necessary to select relevant stratification variables. This requires an advanced level of knowledge about FOSS development. Although some research has been done, the knowledge about FOSS is still in the fledgling stages. This is related to the *second* problem. To produce a strata sample accurate information about the population is required.[572] If for example community affiliation is used as a stratification variable one should know how many percent of the population belong to the free software respectively to the open source community. If this is not known or the knowledge is imprecise one introduces a bias that compensates the advantage of probabilistic sampling. Ghosh et al. (2002b: 2) conclude that there is insufficient empirical data about FOSS developers to identify criteria for sampling. Therefore, it is not possible to demonstrate that a chosen sample of respondents is representative in respect to nationalities, skill levels, income levels, or leadership.

Probability samples and FOSS studies

There is no known FOSS study that is based on a *truly* random sample despite of corresponding claims of the authors of the FLOSS study (Ghosh 2005: 25; Ghosh et al. 2002b: 2). The authors posted the announcement of the survey to different developer forums. Subjects browsing these forums choose themselves to join the survey or not. This method is certainly not random sampling (see also Spector 2001: 15).[573] The argument that the survey was seen “by a very large number of developers” and that “therefore the sample that chose to respond was random” cannot be followed. The developers not motivated filling out surveys have clearly not the same chance filling out the survey as developers highly motivated to do so. In random samples chance selects the subjects included in the survey and not the subjects themselves. Having said this it must be acknowledged that not introducing such biases is very difficult in the FOSS area and that the FLOSS study is in other respects interesting.

Non-probability samples

Besides being the only workable possibility for FOSS research a non-probability sample has the advantage to be less expensive than probability samples. It also may be done spontaneously to take advantage of available respondents.

[572] In many research areas reliable official data is available. For the population of “voters” the real distribution of the stratification variable “gender” is for example known quite exactly. However, for FOSS development there is no reliable official data.

[573] The applied validation procedure does not solve this problem as it compares “verified” (matching of email addresses with names in the source code or in internet archives) with “non-verified” subjects (Ghosh et al. 2002b: 3). Although this approach produces interesting results it adds nothing to the question whether subjects not participating in the survey are distinct from the ones who did. To do so the characteristics of the nonrespondents should have been measured or estimated somehow (Smith 1983: 388). However, Smith (1983: 402) suggests that there is no simple, general, and accurate way of measuring nonresponse bias. Whether a certain method of measuring the nonresponse bias is useful or not depends merely on the content of the survey.

As already mentioned the probability that a subject will be chosen for inclusion in the survey is not known for non-probability samples. In consequence, one cannot claim a priori that the sample is representative for a larger population (Bailey 1994: 94) respectively representativeness does not rest anymore on probability theory but on expert judgment (Deming 1960: 31). While the precision of an estimate about representativeness based on a probability sample is never in doubt the same cannot be claimed for an expert judgment. Deming (1957: 30) lists for non-probability samples the following sampling biases: judgment, selectivity, and availability biases. This limits the ability to generalize beyond the specific sample studied. Most experiments in economics suffer from similar problems.

The problem of representativeness and generalization

Cronbach et al. (1972: 16) call research that intends to generalize a "G study" (G stands for generalizability). If the intention of a research is to establish a causal relationship the ability to generalize is less important. In such a "D study" (D stands for decision) data is collected for the purpose of making a decision or drawing a conclusion. Similarly, Bailey (1994: 94) argue that a non-probability sample is perfectly adequate if the researcher does no intend to generalize the findings beyond the sample as it is for example necessary in the analysis of exit polls in order to predict elections. Ideally, G studies should precede D studies (Cronbach 1972: 18). In some cases, G and D studies are conducted simultaneously. Although not ideal G or D studies may be conducted without their counterpart.

To illustrate the argumentation presented above that the ability to generalize is not always imperative let us suppose an experiment with a non-probability sample. In this experiment the reaction of an experimental stimulus Y on the dependant variable X is researched. Let us suppose that there are two different types of individuals, namely A and B. While in the population 1% of the individuals belong to the A type and 99% to the B type in the sample there are each 50% of A and B individuals. The results show now that for the A subjects X and Y are positively correlated while for B subjects X and Y are negatively correlated. This result is basically independent whether the distribution of A and B in the sample is similar to the one in the population. The causal relationship between the independent and dependent variable is not influenced by the fact that one cannot generalize. However, based on this sample it is neither possible to infer from the sample the distribution of A and B in the population nor whether the correlation between X and Y in the population is in total positive or negative. In the sample the A and B types do equalize themselves. However, because there are almost no A type individuals in the population X and Y are clearly negatively correlated.

The here conducted study does not primarily intend to make inferences about the composition of the FOSS community. [574] A related study would probably exactly research how many of the developers are fun seekers. In fact, this study researches primarily causal relationships such as the ones between different types of motivation, governance structures respectively mechanisms, and development phases. It is for example argued that fun seeking is positively related to autonomy (Hypothesis 9). For researching such questions it is not important whether fun seeking or autonomy are prominently represented in FOSS development or not. The here conducted research is then a D study in which a conclusion is drawn. It seems then to be accountable to

[574] The exceptions are Hypothesis 1 and Hypothesis 2 that are concerned with the importance of different types of motivation. Because this study follows a non-probability sample the related results are not reliable and must be interpreted very carefully.

rely on a non-probability sample. However, a generalization beyond the subjects included in the sample is not appropriate. The suggested hypotheses of this exploratory analysis must then be tested by further research.

Purposive cluster sampling

Out of the different non-probability sampling methods (Bailey 1994: 93-96) quota and dimensional sampling are not suitable because there is, as discussed above, not enough known about FOSS. Snowball sampling is not possible because there seems not to be enough enthusiasm in the population to get an adequate quantity of answers. Contrary, there is some indication that subjects are tired being surveyed.

The sampling method actually executed in this study is purposive cluster sampling. In contrast to normal sampling methods not single subjects are selected from the population but groups or clusters. This sampling method has two main advantages. *Firstly*, it is easy to get access to subjects and *secondly* it is usually cost effective. The big disadvantage is obviously that it cannot be assumed per se that the chosen cluster is representative to the population (but see the related discussion above). It is very likely that the surveyed clusters are biased to some extent. However, despite this disadvantage it is very popular to research clusters. An example is the research by Frey and Meier (2005) to inquire whether economists are more selfish than other people are. The authors used a data set from the University of Zurich. Using this cluster had most probably the advantage that the data was easily accessible. Certainly, the disadvantage is that generalizability beyond students of the University of Zurich cannot be claimed. However, the finding that the significant lower contribution of economists to two social funds of their university is due to self-selection rather than indoctrination by economic theory is interesting in itself. The insight in the related mechanisms is more appealing than the ability to generalize the finding to students from other universities. In the case of this study generalizability would only be important if one were to make concrete predictions about selfishness of students at other universities.

Purposive cluster sampling in this study

As clusters FOSS conferences were selected.[575] FOSS conferences are virtually the only place where developers and users meet physically in a large-scale. This physical presence allows distributing easily surveys on paper.

To get some minimal variation but at the same time not overextending the research budget two conferences were chosen to be surveyed. The FOSS conferences considered to be surveyed were purposively filtered and therefore not chosen randomly. Both conferences had to be hosted in Europe to reduce traveling costs. Additionally, both conferences had to have a minimal magnitude to get a sufficient sample size. Small conferences had therefore no chance of being considered as a cluster. In contrast to these similarities the conferences were selected to have some differences. The two conferences were chosen so that the hosting country[576] and the audience are different.

---

[575] For an overview over different FOSS conferences in Europe see Wikipedia (2006a; 2006c).

[576] Johnson et al. (2005) argue that the survey response style of individuals is related to cultural orientations. An extreme response style that refers to the bias to select more often endpoints of a response scale is more dominant in cultures that rate high on Hofstede's (1980) masculinity and power distance. The acquiescence bias, the tendency to agree with items regardless of its content, is less prevalent in individualistic societies.

9.4 Data collection

There are three basic survey forms for investigating the hypotheses presented in chapter 8: *phone surveys*, *online surveys*, and *paper surveys*. Apparently, phone surveys would not have been applicable because of a lack of corresponding resources and because of a lack of available phone numbers.

The apparently most suitable form for surveying FOSS developers are *online surveys*. FOSS development is almost entirely executed over the Internet. It would be therefore natural to communicate with developers also online. Additionally, online surveys are cheap and easy to handle. There is also no need for data entry which is time consuming and vulnerable to errors. Another aspect is that even within tight budget constraints huge amounts of developers can be contacted. It is therefore not astonishingly that almost all studies about FOSS were conducted online (e.g. Ghosh 2005; Hars & Ou 2002; Hertel, Niedner, & Herrmann 2003; Lakhani & Wolf 2005; Robles et al. 2001a). Initially, it was intended to do so as well and based on the FOSS project phpESP a corresponding online survey was set up.[577] However, in the course of conducting research the intent of using an online survey was abandoned.

In the context of FOSS development there are three main disadvantages of online surveys that are all related to the approach of contacting subjects. One possible approach is to announce the survey in mailing lists of FOSS projects. In many mailing lists such an announcement would be considered as being off-topic. Unless the administrators were to give their approval to post the announcement doing so would hardly be appreciated by the community. Anecdotal evidence suggests that administrators of mailing lists are hesitant to give approval for such postings because many FOSS developers have already participated in various FOSS studies. Another approach is to ask a FOSS hosting platform for support. Because a certain population size is needed to get enough answers, only SourceForge and few other hosting platforms were possible partners. However, feedback from other researchers suggested that it is difficult to get such support. Yet another approach used by some researchers is to harvest email addresses with a web crawler and contact the developers by email. Three problems are associated with such a proceeding:

- *Firstly*, it can be assumed that the responding rate to such emails is very low. Such emails are unsolicited which qualifies them clearly as spam or unsolicited bulk email. Unsolicited requests for survey participation sent over the Internet are considered to be more annoying than requests delivered by postal service. The reason for this difference is explained in the RFC 1855 of the Internet Engineering Task Force (IETF) on Netiquette (Network Working Group & Hambridge 1995: 4):

  > *The cost of delivering an e-mail message is, on the average, paid about equally by the sender and the recipient (or their organizations). This is unlike other media such as physical mail, telephone, TV, or radio. ... This is a fundamental economic reason why unsolicited e-mail advertising is unwelcome (and is forbidden in many contexts).*

  Unsolicited email is therefore an ethical problem and should be avoided.

- *Secondly*, based on the negative touch of unsolicited bulk email a major bias concerning subjects filling out the survey must be expected. Only subjects with high tolerance concerning Netiquette were to participate.

[577] For an interesting description of different FOSS applications to conduct online surveys see GESIS (2007, only in German).

- *Thirdly*, anecdotal evidence suggests that FOSS developers are over-surveyed. Huge amounts of students as well as researchers conduct online studies about FOSS. This is facilitated because of the marginal costs conducting a survey. Feller and Fitzgerald (2002: 2) state that „the recent surge of academic inquiry into OSS is not at all surprising – the global research community has, in fact, been jumping onto a bandwagon that was already quite crowded". In consequence, the return rate is thought to be small. Relying on a method of data collection nobody else has used before seems to be an interesting alternative to online surveys.

Certainly, an alternative to contact developers online would have been to contact them in person. At the FOSS conferences participants could have been motivated to participate and given a flyer with the URL of the online survey. However, it was assumed that not all developers have packed their notebook if the stroll around at a FOSS conference. [578] While hackers are expected to carry likely a notebook this does not hold true for other developers such as freshmen. In consequence, a bias would have been introduced.[579]

Overall it can be summarized that there are weightily advantages of online surveys (e.g. costs) but that there are also clear disadvantages (ethics, biases, and return rate).

An alternative to an online survey is a survey conducted on *paper form*. The advantage of a paper form survey is that it is not considered of being spam. It can also be expected that due to the very unusual form in FOSS research the response rate is higher and that the responses are less biased. The disadvantages of paper surveys are on the one hand the costs (e.g. postage) and on the other hand getting in touch with the developers. The former problem can be reduced by surveying clusters, the later by conducting the research at FOSS conferences (see chapter 9.3). These conferences are the only opportunity to meet a wide spectrum of developers as a big cluster physically. To meet developers at conferences and ask for survey participation has the additional advantage to learn more about the surveyed cluster and assessing the bias based on subjects not participating in the survey.

9.5 Item scaling

There are many different methods of item[580] scaling such as the Likert scale (Likert 1932), the Thurstone scale (Thurstone 1927, 1928; Thurstone & Chave 1929), the Guttman scale (Guttman 1950, cited in Bortz and Döring 1995: 204), the Edwards-Kilpatrick scale (Edwards & Kilpatrick 1948), the Rasch scale (Rasch 1980), or the Coombs scale (Coombs 1948, 1966). [581]

The Thurstone and Edwards-Kilpatrick scale require good prior knowledge that is not thought to exist in the case of this study (see Bortz & Döring 1995: 202-210). Rasch, Guttman, and Coombs scales seem to be too demanding to develop and too laborious for this doctoral thesis. However, the Likert scale seems to be an appropriate scale. The Likert scale has the advantage that is simple and that it yields, compared to other scales

[578] In fact, at FOSDEM almost all developers packed a notebook. However, the wireless connection was not working properly so that it would have been difficult for them to fill out the online survey. At the Chemnitzer Linux-Tage many but by far not all developers carried a notebook with them.

[579] It turned out to be true that conducting an online survey would have introduced a major bias towards hackers.

[580] Items are quantified records (adapted from Edwards & Bagozzi 2000: 156).

[581] For reflexive constructs *scales* are built while for formative constructs *indexes* are built (Diamantopoulos & Siguaw 2006: 264; Diamantopoulos & Winklhofer 2001: 269). Because in this study a mixed model is applied both terms are used interchangeably.

such as the Thurstone scale, a high reliability with few items (Likert 1932). A huge advantage is that many studies related to motivation (e.g. Deci & Ryan 2006) and FOSS (e.g. Hertel, Niedner, & Herrmann 2003) have already used Likert scales that makes it possible to reuse corresponding items. For this reasoning a five-point Likert scale is thought to be appropriate for this study.

Likert scale

In a Likert scale subjects are asked to specify their level of agreement to a list of items. The response options are bipolar ranging from “strongly disagree” (1) to “strongly agree” (5). This method of inquiring is based on the assumption that attitudes are not an inflexible and rigid element of personality but within a range in which responses move (Likert 1932: 8). The original Likert scale has five points what is adopted here too (Likert 1932: 14). A three-point scale is inappropriate because too much information is lost which compromises power (Baroudi & Orlikowski 1989: 99). Seven-point scales are not very convenient for subjects conducting the survey because they are too laborious to answer.

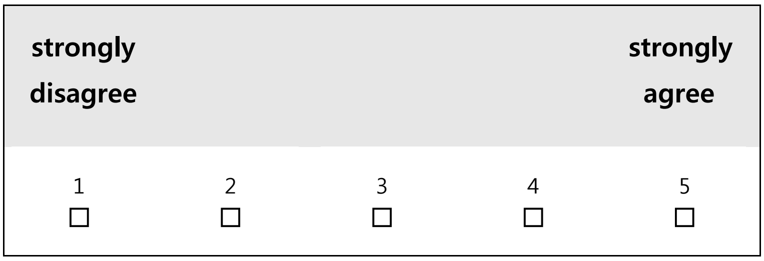


Figure 103: Answer options in this survey[582]

In contrast to the original scale of Likert (1932: 14) only the extreme values are named. To name the answer option in the middle (3) with “undecided” has several shortcomings. *Firstly*, it invites not thinking about the item by quickly answering undecided. This results in low item discrimination (see chapter 11.2). *Secondly*, it is theoretically challenging to interpret the answer “undecided”. The middle position cannot be interpreted as the arithmetic middle position between the two extremes. In consequence, computing certain statistical procedures do not seem to be appropriate.

It is common for complex concepts that they are made up of several different items. The score of each item of a concept is summed up (Likert 1932: 26). In this study the sum of the different items measuring the same concept was divided by the number of items. This ensures that concepts with a different number of items are, at least to some extent, comparable.

Level of measurement

The level of measurement of the test scale is clearly ordinal.[583] The answer categories from “strongly disagree” to “strongly agree” do have an inherent order and values can be ordered as “greater” or “less” (see Stevens 1946: 678-679). The scales are not on an interval level because differences of intervals are not equal.

[582] In the survey the values of the answer options (1-5) are not shown.

[583] In a now widely adopted typology of scale measurement Stevens (1946) proposes to distinguish between nominal, ordinal, interval, and ration scales (for a critique of this typology see Velleman & Wilkinson 1993).

Permissible statistics for ordinal scales include medians and modes but not means.[584] For ordinal scales monotonic transformations are permissible which do not change median and mode but the mean (Aguinis, Henle, & Ostroff 2001: 29). In addition, percentile ranks, correlation coefficients based on ranks such as Spearman's rho or Kendall's W, and rank-order analysis of variance can be used.

9.6 Measurement model specification

For researching complex constructs multiple measures are often used (MacKenzie, Podsakoff, & Jarvis 2005: 711). The reason for this is that i) constructs can hardly be measured without error, that ii) it is hard for a single indicator to adequately capture the breadth of a construct, and that iii) it is crucial to unconfound the measurement method from the construct of interest.

However, using multiple measures causes the problem to specify the relationship between the measures and the construct of interest (MacKenzie, Podsakoff, & Jarvis 2005: 711). There are two different types of generic measurement models dealing with this problem: the common latent construct model with reflective indicators and the composite latent construct model with formative indicators. In the following, these two models and the corresponding specifications are discussed.

Reflective constructs

Classical test theory assumes that the variation in the scores of the items of a construct is a function of a true score plus error (Jarvis, MacKenzie, & Podsakoff 2003: 199). The latent construct thus causes the observed variation in the items (2003: 200) and item values are effects of the latent variable. The assumed direction of causality in such a *reflective*[585] model is from the latent variable to its items (see Figure 104). Changes in the latent variable cause changes in the items. The items should be internally consistent and interchangeable. Removing a valid item should then not have an influence. This is because each item is assumed equally valid. In consequence, the items should correlate with each other. For example if satisfaction with a car (latent construction) is inter alia measured by the items “I appreciate my car” and “I am happy with my car” these two items should correlate with each other because they measure the same thing. If an item does not correlate with other items respectively if it shows no internal reliability, measured for example by Cronbach's alpha (Cronbach 1951), one should remove it (Diamantopoulos & Siguaw 2006: 267). A reliability coefficient demonstrates whether “a certain collection of items … yield interpretable statements about individual differences” (Cronbach 1951: 297).

[584] This holds only true for items that use a Likert scale.

[585] The naming of the term *reflective* is based on the fact that the different items in such a model reflect a latent variable (Edwards & Bagozzi 2000: 155).

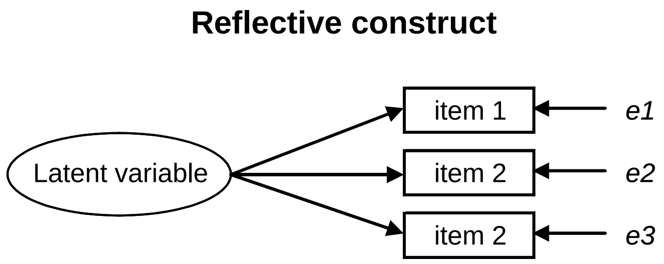


Figure 104: Simplified model of a reflective construct

Formative constructs

In some instances the direction of causality may be as specified above. However, this is not always the case. For some constructs it is more reasonable to view the causality flowing form the items to the constructs (see Figure 105). Changes in the items cause changes in the latent variable. An example is socioeconomic status that is formed as a combination of education, income, occupation, and residence (Diamantopoulos & Winklhofer 2001: 270). Such a model is called *formative.*[586] It is assumed that every single item has an important impact on the construct (Jarvis, MacKenzie, & Podsakoff 2003: 201-202). The items jointly determine the empirical meaning of the latent variable. Items may be correlated but it is not necessary. It is possible that individuals have a high income but a moderate education and vice versa. The latent variable may be represented by mutually exclusive types of behavior. Internal consistency is then not needed and items that are not correlated to other items should not be removed. The error term for formative models is, unlike for reflective models, not at the item but at the construct level.

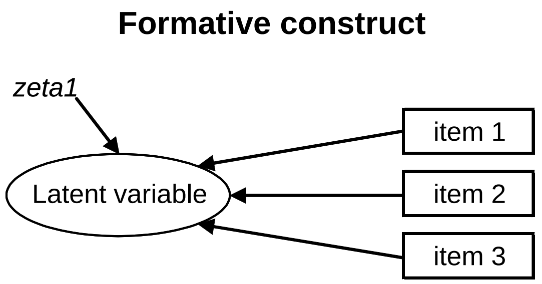


Figure 105: Simplified model of a formative construct

Reflective vs. formative constructs

There not only differences between reflective and formative constructs as described above but also similarities (Jarvis, MacKenzie, & Podsakoff 2003: 202). Both reflective and formative models possess a surplus meaning not captured by the items used to measure them. In both models the scale scores do not adequately represent the construct because there are always measurement errors.

Jarvis et al. (2003: 201) sum up the differences and similarities as follows:

[586] The naming of the term formative is based on the fact that the latent variable is formed by different items (Edwards & Bagozzi 2000: 155-156).

Reflective model	Formative model
• Direction of causality is from construct to item	• Direction of causality is from item to construct
• Items are expected to be correlated, items should possess internal consistency respectively high reliability	• No reason to expect that the items are correlated; internal consistency is not necessary
• Dropping an item from the model does not alter the meaning of the construct	• Dropping an item from the model may alter the meaning of the construct
• Measurement error is taken into account at the item level	• Measurement error is taken into account at the construct level
• Construct possesses surplus meaning	
• Scale score does not completely represent the construct	

Reflective vs. formative constructs: An example

An example adapted from Albers and Hildebrandt (2006: 11-12) may show the differences between reflective and formative constructs (see Figure 106). If somebody is satisfied with his hotel accommodation he may state that he appreciates the hotel and that he feels comfortable. These two items should be highly correlated. In most cases an individual who appreciates a hotel does also feel comfortable.

However, satisfaction with the hotel accommodation may also be measured with the degree the individual consider the hotel staff friendly and how good he evaluates the wellness area (see Figure 106). While these two items of this formative construct measure both satisfaction with the hotel accommodation they are not necessarily correlated. It may be that the hotel staff is very friendly but that the wellness area is poor. The overall satisfaction is then measured by adding the score attained by the wellness area and the score attained by staff friendliness. Deleting the item that measures staff friendliness may totally change the measured satisfaction with hotel accommodation.

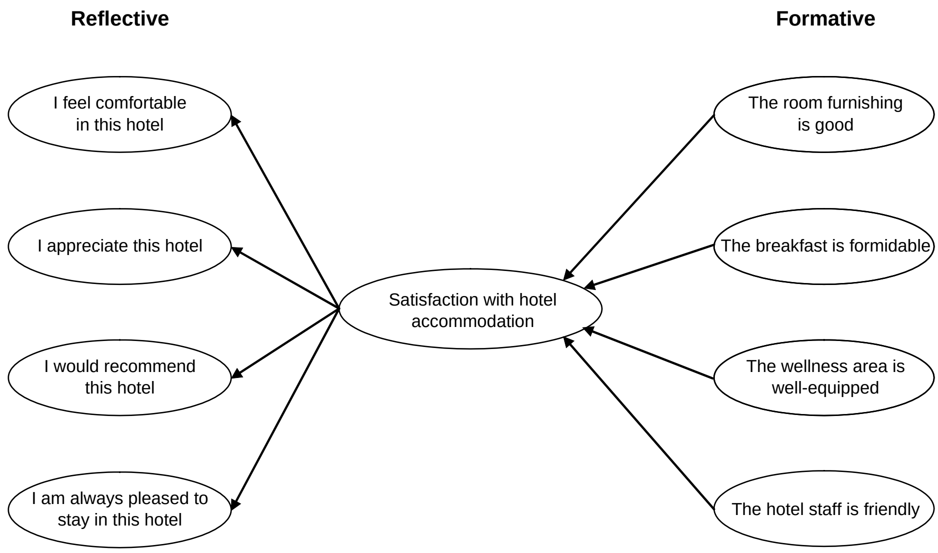


Figure 106: Satisfaction with hotel accommodation[587]

This example shows that reflective as well as formative constructs have its advantages:

- The overall satisfaction is probably easier measured with a reflective construct. The satisfaction can be surveyed with a few good items. Adding more items should not improve the overall score because all items are highly correlated. For a formative construct the overall satisfaction depends heavily on whether all possible factors influencing satisfaction are included or not. Therefore, it may be necessary to include many items.
- The problem with reflective constructs is, however, that it does not indicate why a hotel guest is satisfied or not. In contrast, a formative construct tells the hotel manager more about the structure of customer satisfaction. If the hotel scores always high on staff friendliness, wellness area, and breakfast but very low on room furnishing it seems to be obvious how customer satisfaction can be increased.

In a mixed model the advantages of both constructs may be combined.

Mixed reflective and formative constructs

There are models that include multiple formative and reflective dimensions (Jarvis, MacKenzie, & Podsakoff 2003: 204; MacKenzie, Podsakoff, & Jarvis 2005: 713). A first-order construct may consist of reflexive items but may be itself a formative item of a second-order construct (see Figure 107). It may also be that some of the first-order dimensions are formative items and others reflective items. MacKenzie et al. (2005: 713) state:

> *… it is possible for a single multidimensional construct to have one type of measurement model relating its measures to its first-order subdimensions and a different measurement model relating its subdimensions to the second-order latent construct they represent. It is also possible for a construct to have a mixture of some reflective and some formative indicators at either level of abstraction.*

[587] Adapted from Albers and Hildebrandt (2006: 12).

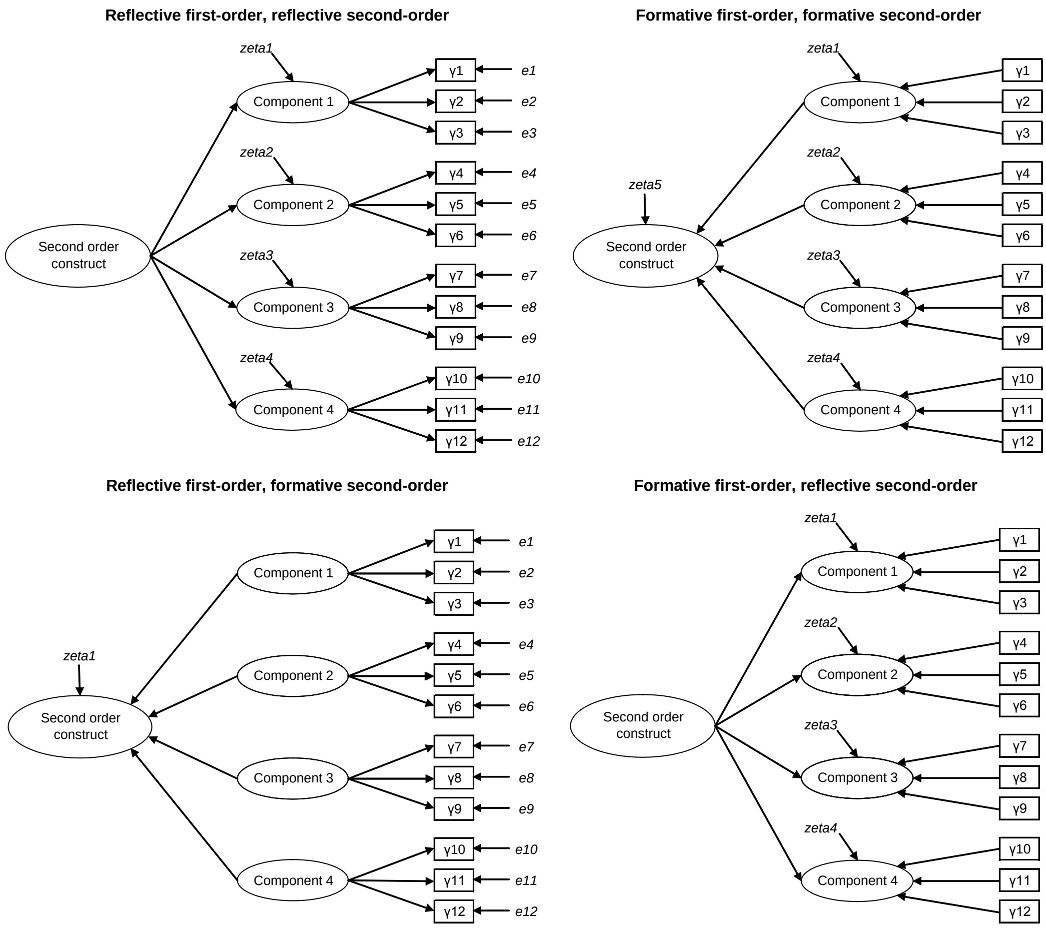


Figure 107: Alternative first and second order constructs[588]

It may even be more complex because the same item may be both of reflective respectively formative nature (Fayers et al. 1997). The directionality of the relationship between an item and a construct may be far from obvious (Fayers et al. 1997: 395). Insomnia decreases the quality of life. Insomnia is then part of a formative construct because it *causes* a loss of quality of life. Quality of life depends also on other factors totally unrelated and not correlated to insomnia. However, it is also may be that insomnia is part of a reflective construct because it is the *effect* of poor quality of life.

Reflective vs. formative constructs: The problem of a wrong specification

Research including latent items adopts usually a reflective model (Diamantopoulos & Siguaw 2006: 263; Edwards & Bagozzi 2000: 155; Fayers et al. 1997: 405; Jarvis, MacKenzie, & Podsakoff 2003: 200). A study of the top four marketing journals found that 96% of all models are reflective (Jarvis, MacKenzie, & Podsakoff 2003: 206). However, more than one fourth of these models are wrongly specified. The wrongly specified models should have used a formative construct. In consequence, items have wrongly been deleted because of their low reliability (e.g. Cronbach's alpha).

A wrongly specified model may cause major errors. This can be pointed out by a fictional study on alcoholism. In such a study one may ask how much beer, wine, and liquor is consumed. A proper model is formative

[588] Jarvis et al. (2003: 205)

because one may suffer from alcoholism by drinking only hard liquor but no beer or wine.[589] Let us assume that most consumers who prefer liquor usually drink only this kind of alcohol but hardly beer or wine. Let us further assume that wine drinkers consume usually also beer and liquor and that beer drinkers consume wine and liquor as well. In consequence, wine and beer consumption are highly correlated with each other while there are no major correlations between liquor and beer respectively wine consumption. If this study were wrongly specified as a reflective construct the item concerning liquor consumption would be excluded because of a low Cronbach's alpha. Thus, a person drinking two bottles of liquor a day is wrongly categorized as being abstinent. According to Jarvis et al. (2003: 207) such an inappropriate application of a reflective measurement model is by far the most common type of specification error in the measurement model.

Deciding: reflective or formative?

As argued above a correct model specification seems to be very important. However, the determination whether a construct is formative or reflective is not trivial (Diamantopoulos & Siguaw 2006: 265; Hulland 1999: 201). Establishing the causal relationship between a latent variable and its items is difficult and therefore it is not easy to choose between a reflective, a formative, or a mixed model (Bollen & Ting 2000: 4; Edwards & Bagozzi 2000; Fayers et al. 1997: 395; Hulland 1999: 201). To ease the decision between formative and reflective models Jarvis et al. (2003: 203) provide the following assistance:

Topic	**Items**
Direction of causality	• Are the items defining characteristics or manifestations of the construct? • Would changes in the items cause changes in the construct or not? • Would changes in the construct cause changes in the items?
Interchangeability of items	• Should the items have the same or similar content? Do the items share a common theme? • Would dropping one of the items alter the conceptual domain of the construct?
Covariance among the items	• Should a change in one of the items be associated with changes in the other items?
Nomological net of the construct items	• Are the items expected to have the same antecedents and consequences?

Figure 108: Decision support of model specification

Fayers et al. (1997: 396) report a solution for choosing between formal and reflective models. Based on Hume's (1739) discussion on causality temporal priority is a condition of causality. If X precedes Y it must also cause Y. However, although a cause precedes an effect it is sometimes difficult or impractical to find out what

[589] In fact, 22% of individuals who suffer from alcoholism consume solely one type of alcohol (Klatsky, Armstrong, & Kipp 1990).

precedes what. This refers to the old problem "which was first, the bird or the egg" (Plutarch 1878: 242)?[590] The bird emerges from an egg but the egg is laid by a bird. It is therefore not clear which gave rise to the other.

9.7 Axioms of classical test theory

Based on the axioms of classical test theory (Novick 1966) a test should meet three criteria (Lienert & Raatz 1998: 7; Rost 2004: 33): Objectivity[591], reliability, and validity.

9.7.1 Objectivity

A test can be considered objective if "every observer of a performance arrives at the same report" (Cronbach 1990: 33). Results of a test should then not arise from subjective interpretations. An essay test is objective if independent judges assign completely the same scores.

Tests are completely objective if the subjects choose the best answer themselves such as for example in multiple-choice tests (Bortz & Döring 1995: 181; Cronbach 1990: 33). In such tests the answers cannot be interpreted differently than intended by the subject.[592/593]

9.7.2 Reliability

In classical test theory reliability refers to the extent to which test scores are free of errors from measurement (American Psychological Association et al. 1985: 19). A reliable score is measuring constantly and accurately. Reliability is important because a person filling out a survey may perform differently in one occasion than in another for reasons not associated to the purpose of the study. A person may be tired, more familiar with the items, or just guessing more in one occasion than in another. If so, differences between items of a score or between similar scores may be attributable to an error of measurement.

There are different methods for testing reliability (Aguinis, Henle, & Ostroff 2001: 33-36; Bailey 1994: 72-73; Rost 2004: 377-382). Multiple-forms reliability is based on two surveys each using different items but designed to measure the same concept. If a test is reliable the scores of the two surveys should correlate heavily. For Split-half reliability twice as many items as a researcher needs are included in the survey. A score based on the first half of the items should correlate with the second half of the items. A slightly different approach is retest reliability. A score is reliable if the results in an original survey and the results in the repetition (with the same subjects) are similar.[594]

---

590 Certainly, the question should not be taken literally because biologically neither the bird nor the egg was first. Because species evolve evolutionary there was no first egg or no first bird.

591 Aiken (1985: 88) classifies objectivity as interscorer or interrater reliability. In consequence objectivity is a special version of reliability.

592 Certainly, there are marginal exceptions. For example a cross may be ambiguous so that it must be interpreted.

593 Of course, this holds only true if malicious manipulation is taken into account.

594 Retest reliability can only be used if there is no change in the researched phenomenon (Bailey 1994: 73). Whether this is the case or not depends on the time frame and on the characteristics of the concept being surveyed.

Cronbach alpha

The most known and used method for assessing reliability is Cronbach's alpha (Cronbach 1951: 330).[595] Technically it is the "average of all possible split-half coefficients for a given test" (Cronbach 1951: 300). It measures internal consistency respectively the degree to which items of a measure are correlated with each other (Aguinis, Henle, & Ostroff 2001: 34-35). It is generally thought that Cronbach alpha underestimates reliability (Bortz & Döring 1995: 184; Cronbach 1951: 308).

Cronbach alpha can take values between negative infinity and one although only positive values can be interpreted. The higher alpha is the better is the internal consistency and reliability. There are different proposals for minimal levels of acceptable reliability (Charter 2003: 291-292). While Heiman (1995: 283) considers a reliability higher than 0.80 as good reliability the same rating is given by Weise (1975: 219) only for a reliability higher than 0.90. Another suggestion is to judge internal consistency below 0.70 as unacceptable, between 0.70 and 0.79 as fair, between 0.80 and 0.89 as good, and above 0.90 as excellent (Cicchetti 1994: 287). Similarly, for Litwin (1995: 31) a reliability of 0.70 is sufficient. Acceptable levels of reliability vary between different fields of application (Charter 2003: 295) and they depend on the purpose of the study (Aguinis, Henle, & Ostroff 2001: 36). For example the reliability in intelligence tests is generally higher than in tests on personality. For Aquinis (2001: 36) a reliability of 0.70 is in most cases sufficient for measures in work and organizational psychology. He further states that a reliability even below 0.70 may be acceptable for research purposes.

For interpreting reliability coefficients it is important to reflect on possible sources of measurement error:

- ***Difficulty of topic:*** It seems to be quite clear that reliability is related to the topic of a study. If the topic is difficult to grasp a low reliability must be expected. Items easy to answer such as for example the age of subjects should be answered more constantly and accurately than an item about the motivation to contribute to a FOSS project.
- ***Accuracy of items:*** To achieve high reliability the items have to be very clear and accurate. Questions with such characteristics must either be heavily pre-tested or taken from a standardized set of questions tested by other researchers. The former possibility has the disadvantage that it needs scare resources. The later possibility may suffer from the matter of fact that the standardized items are not adjusted to the specific requirement of the study.
- ***Language proficiency:*** If subjects have a low proficiency in the survey language one may suppose a lower reliability.
- ***Number of items:*** The number of items to measure a latent variable influences Cronbach alpha. Cronbach alpha is generally the higher the more items are used (Cronbach 1951: 330). This can be compared to the following principle: "When a message is repeated over and over, it is easier to infer the true message even when there is substantial interference (item unreliability)" (Cronbach 1951: 330).[596] Superficially, this may be interpreted to add additional items and thereby to lengthen a survey

[595] Interestingly, Cronbach (2004: 3) himself "doubt whether coefficient alpha is the best way of judging reliability". He concludes that the "choice of a single statistic to summarize the accuracy of an instrument is not the best report that can be made. I recommend that the three separate components of variance be reported" (2004: 27).

[596] The same relationship between the number of items and reliability holds true for other methods to measure reliability such as the Kuder-Richardson Formula 20 that is similar to Cronbach's alpha except that it is used for dichotomous items (Ebel 1967: 125).

(see also Rost 2004: 383-384). However, Cronbach (1951: 330) argues that doing so is not without disadvantages because this leads to redundant information. The more a message is repeated the less time there is for other relevant information. By including additional items to measure the same concept internal consistency can be increased but the scope of the study is compromised. Including more items may also lead to symptoms of fatigue, decreased motivation to finish the survey, and considerable irritation (Rost 2004: 383, see also below). A score with twenty or more very similar items may bore subjects.

A problem if reliability is maximized may be that validity is minimized (Rost 2004: 392-394). According to the attenuation paradox (see Loevinger 1954) increasing reliability may lead to the problem that something irrelevant or wrong is measured *very precisely*. To choose items that lead to high reliability has the consequence that items get more and more similar. However, real-life phenomena in social science are complex and hardly completely unidimensional. By increasing reliability to very high levels one force a scale artificially into unidimensionality. In consequence, the validity of the score decreases.

After all, it can be concluded that for the purpose of this study a reliability above or just below 0.70 is sufficient. Certainly, a higher reliability would be desirable. However, this would also have pitfalls.

9.7.3 Validity

The validity of a score is defined as the extent to which it is measuring accurately the concept in question and not another concept (Aguinis, Henle, & Ostroff 2001: 37-41; Bailey 1994: 67). Validity is therefore an inquiry into the soundness of an interpretation (Cronbach 1990: 150-151). Generally, to measure validity is more difficult than to measure objectivity or validity (Litwin 1995: 44).

Reliability does not imply validity. A reliable questionnaire may measure something accurately and constantly but not inevitably what it is supposed to be measuring (see also the attenuation paradox in chapter 9.7.2). A male calibrated thermometer may be reliable but not valid. It measures the temperature highly constant but at the same time very wrongly.

According to Cronbach (1990: 151) there are three types of validity (see also Aguinis, Henle, & Ostroff 2001: 38-41):[597]

- ***Content validity*** is merely assessed by researchers studying the concept to be measured and determining in the best judgment whether the instrument arrives at the concept sufficiently (Bailey 1994: 69). Such a judgment is partially definitional or semantic. In consequence, there is some concern about the usefulness of this approach (Litwin 1995: 35). However, it must be acknowledged that due to resource scarcity this may be the only possibility for test validation.
- ***Criterion validity*** refers to a comparison of a test with another test known to be valid (Bailey 1994: 70). The reasons to replace an old valid test by a new one are manifold (1994: 70). The old test may be complicated to use, consist of too many item, difficult to code, or the language may be outmoded. A variant of criterion validity is predictive validity (Cronbach & Meehl 1955: 281). Hereby the validity

[597] Content validity is also known as logical or face validity (Bailey 1994: 68). Criterion validity is also labeled pragmatic validity (Bailey 1994: 68-69).

of a test is approved by its ability to predict future events. If such a prediction is possible it is surely a very strong method of validation.

- ***Construct validity*** is "a matter of asking though questions about the test content and its correlates" (Cronbach 1990: 160). To do so different hypotheses about the test should be set up. If the test is valid and its theoretical basement is sound the data collected with the test should lead to the refusal of the null hypotheses (Bortz & Döring 1995: 186). For example a qualification test for prospective fire fighters may measure vocabulary and reading ability because they are assumed to heavily influence training success (Cronbach 1990: 159). The question that should be asked then promptly is whether the training is unnecessarily based on verbal demands. If so, fire fighters with higher test scores were to have only higher training success if course instructors would teach overly complicated. If construct validity is tested one should attempt to get rid of *unnecessary* linguistic complexity. If training success is still related to vocabulary and reading ability the test is valid. Otherwise the test does not measure the qualification of being a fire fighter but to cope with linguistic complexity.

9.8 Parametric vs. non-parametric tests

For inferential statistics the most powerful test should be used. Generally, parametric tests are more powerful than non-parametric tests (Baroudi & Orlikowski 1989: 98).[598] Parametric tests assume special population distributions and conditions. For example F- and t-tests assume a normal distribution in the population and equal variances in the tested groups. If these assumptions are violated one should rely on non-parametric tests because they have certain desirable properties that hold under mild assumptions regarding the underlying populations (Hollander & Wolfe 1999: 1). For example most non-parametric tests are distribution free respectively do not assume that the population is somehow distributed (e.g. normal distribution). Therefore such non-parametric tests are called to be more robust (Gibbons 1985: 27; Maritz 1995: 1). Because of this many non-parametric tests are still appropriate if "dirty-data" is used (Gibbons 1985: 25). Gibbons (1985: 24) concludes that by using non-parametric statistics one does get less likely "into trouble" or misuse statistics.

Depending on the measurement level it is not always possible to use every statistical test (Stevens 1946: 678). While parametric tests rely on interval or ratio scales non-parametric tests rely at least either on nominal (e.g. Chi-square test) or on ordinal scales (e.g. Kruskal-Wallis test).[599] Another huge advantage of non-parametric statistics is that they are i) easier to apply and ii) easier to understand (Hollander & Wolfe 1999: 1).[600]

[598] Power is defined as the probability of rejecting the null-hypothesis if it is false (Gibbons 1985: 26). See also chapter 9.2.

[599] See also footnote 583 and chapter 9.8.

[600] However, even for very simple statistics unexpected problems arise. For example contingency tables used to analyze categorical variables have pitfalls as Simpson (1951) showed impressively. What is now known as the Simpson paradox is not only of theoretical but also of practical relevance (e.g. see the success rates of two treatments for kidney stones Julious & Mullee 1994). A Simpson's paradox occurs if the successes of groups seem to be reversed if the groups are combined. A known incident is the Berkeley sex bias case (Bickel, Hammel, & Oconnell 1975; Maxwell & Jones 1976). The University of California at Berkeley was blamed for having a bias against women applying to graduate school. In fall 1973 men (44% admitted) applying to graduate school were more likely admitted than women (35% admitted). The difference was so large that chance was clearly ruled out as an explanation (there were 12'763 valid applications). However, an analysis found that no single department was significantly biased against women. Actually, the data revealed a significant bias against men. How can this be explained? Women applied more frequently to competitive departments (for either sex) with low admission rates while men applied more often to less-competitive departments with high admission rates. Consequently, although women had in almost every single department a higher admission rate the overall admission rate was lower than for men.

While in psychology non-parametric as well as parametric tests are used economics has been almost exclusively devoted to parametric statistics. A notable exception is the economist Milton Friedman (1937: 675) who claimed that the assumptions necessary for the valid application of analysis of variance are often not justified. He further argues that for social and economic data normal distribution is likely the exception and not the rule. In consequence, he developed the non-parametric Friedman test that is similar to the more known Kruskal-Wallis one-way analysis of variance (see also Kruskal & Wallis 1952: 601-602).

Using parametric tests although assumptions are violated?

Labovitz (1967: 151) argues that certain assumptions of parametric tests can be safely violated (see also Gardner 1975). He argues for example that t-tests and F-tests are robust against the violation of the assumption of normal distribution and against the violation of variance homogeneity (Labovitz 1967: 156-158).[601] Labovitz (1967: 158) shows that in some cases the violation of an assumption leads to *lower* significance. Thereby the introduced error lowers the chance of rejecting the null-hypothesis that is methodologically no problem.

Ordinal test scales are frequently treated as interval test scales (Knapp 1990; Rost 2004: 65). Labovitz (1967: 152 and 159) argues that a strict adherence to measurement scales may lead to an extensive waste of information and that treating an ordinal scale as an interval scale results only in small errors. Stevens (1951: 26, cited in Gardner 1975: 52) observes similarly that “in the strictest propriety the ordinary statistics involving means and standard deviations ought not to be used with these scales {ordinal scales}” but that “for this 'illegal' statisticizing there can be invoked a kind of pragmatic sanction: in numerous instances it leads to fruitful results”.

In consequence, Cohen (1965: , cited in Baroudi and Orlikowski 1989: 1998) argues that a parametric test should only be substituted by a non-parametric test if there is an extreme violation of assumptions that rarely occurs in behavioral research. Baroudi and Orlikowski (1989: 98) conclude that “researchers are encouraged to use the parametric test most appropriate for their study and resort to non-parametric procedures only in the rare case of extreme assumption violations”.

Using non-parametric tests although the power is lower?

Hollander and Wolfe (1999: 1) argue that non-parametric tests are only slightly less efficient than parametric tests if the population is normally distributed. Gibbons (1985: 30) reports that the relative efficiency of the Spearman rank respectively the Kendall Tau-b correlation compared to the Pearson product-moment correlation is 0.912. The relative efficiency of the Kruskal-Wallis test compared to the F-test is 0.955. Yet, if the population is not normally distributed non-parametric tests are mildly or widely more efficient.

[601] However, Labovitz (1967: 158) argues also that the robustness of parametric tests decreases sharply if several different assumptions are violated *simultaneously*.

Parametric or non-parametric tests?

It can be concluded that:

i. If the assumptions about the population are satisfied, parametric tests should be used.

ii. If the assumptions about the population are not satisfied, non-parametric tests should be used.

iii. In either case parametric and non-parametric can be used if some loss of power is accepted.

The tests used in this study follow this conclusion. As non-parametric tests chi-square tests, Kruskal-Wallis tests, and Kendall Tau-b correlations[602] will be used.

9.9 Summing up

The basic insights of this chapter are:

- It was discussed why a cross sectional study design and not a longitudinal study design has been chosen. Similarly, an explanation was given why the data of this study is collected by a survey and not by observation. The problem of distinguishing between period, age, and cohort effects was noted. > See chapter 9.1
- The four elements of statistical inference (sample size, significance criterion, population effect size, and statistical power) were discussed. It was analyzed why to look only at the significance level is highly problematic. These four elements were set for this study and the corresponding tradeoffs were discussed. The target sample size of the survey was computed to be 200 subjects. > See chapter 9.2
- As a sampling method purposive cluster sampling was chosen. Because this is a non-probabilistic sampling method hypotheses concerning population characteristics are not reliable. However, hypotheses about causal relationships can be still tested reliable. > See chapter 9.3
- As clusters two FOSS conferences were selected. > See chapter 9.3
- To measure concepts five-points Likert scales are used. The middle position is purposively not named. > See chapter 9.5
- The differences between formative and reflective models were presented. > See chapter 9.6
- The three axioms of classical test theory objectivity, reliability, and validity were discussed. > See chapter 9.7
- It was specified under which circumstance parametric respectively non-parametric tests are used in this study. > See chapter 9.8

This chapter discusses the methodological and statistical assumptions. It sets thereby the methodological and statistical frame for hypotheses testing. Two aspects of this chapter are crucial because they limit the subsequent empirical interpretation: *Firstly*, a cross sectional and not a longitudinal study is conducted. This causes the problem that it cannot be ruled out that found relationships between motivation and development

[602] A Kendall Tau-b correlation is preferred to a Spearman rank correlation because the former is more reliable, more conservative, and better to interpret (Gibbons 1985: 297).

phases are influenced by period or cohort effects. *Secondly*, a non-probabilistic sampling method has been chosen. This clearly impedes the ability for generalization. However, this does not restrict the ability to research causal relationships. In FOSS research basically all studies suffer from these two limitations.

10 The Survey: Data for hypotheses testing

In this chapter the surveyed clusters, the survey, the used items, and data handling are discussed.

10.1 Surveyed clusters

The survey was distributed on paper at two FOSS conferences: The FOSDEM and the Chemnitzer Linux-Tage. These two conferences will be briefly described in turn.

FOSDEM

The FOSDEM conference took place in Brussels (Belgium) at the campus of the Université Libre de Brussels at the 25th and 26th of February 2006. The whole conference was free but many participants donated money. The language spoken among the participants as well as the conference language was English although French is usually spoken Brussels. During the two days of the conference the surveys were distributed at the info desk. Conference participants asking for a free information package including schedule, map, speech description, and some promotional material were asked concurrently to join the survey.

The oral feedback concerning the survey was very good. It was particularly often mentioned that the survey items are suitable for the FOSS community. However, there were some comments indicating resistance against survey participation in general like “we are over-surveyed” or “not yet another survey”. A major motivation for participation seemed to be to help a student finishing his Ph.D. The assumption that university sponsorship of studies increases response rates (Jones 1979: 103) seems to hold true for this study. In consequence, the instructions for the subsequent Chemnitzer Linux-Tage conference were complemented by the stating that the data of the study are needed for a Ph.D.

Subjectively, many participants were experienced developers and deeply rooted in the FOSS and especially in the free software community.

Chemnitzer Linux-Tage

From the 04th to the 5th of March 2006 the surveys were distributed at the Chemnitzer Linux-Tage. This conference took place on the campus of the Chemnitz University of Technology in Germany. It is one of the oldest and biggest FOSS conferences in a German speaking country. The entrance fee for two days was 5 euro for regular participants; the reduced fee was 3 euro. Contrary to the FOSEDM conference the presentations were given in German.

The conference organizers provided a booth with a poster in A0 size informing about the study and a table with several chairs. This provided the opportunity for participants to fill out the survey immediately at the booth. The booth was located next to the entrance. Conference visitors who came to pass the booth were kindly asked to fill out the survey. Additionally, the staff of other booths was actively asked to participate.

The oral feedback received at the Chemnitzer Linux-Tage from the visitors stated that the items were demanding to answer. It was for example several times asked what a fork is. The booth staff gave about the same (positive) feedback as the one received at FOSDEM.

Subjectively, many participants apart from the booth staff were students just starting being part of the FOSS community. The majority of *visitors* did not seem to make a difference between the open source and the free software community.

FOSDEM and Chemnitzer Linux-Tage compared

There seem to be *subjectively* three major differences between FOSDEM and the Chemnitzer Linux-Tage:

i. The survey at the FOSDEM conference was considered appropriate while visitors at the Chemnitzer Linux-Tage considered it demanding.

ii. The visitors of the Chemnitzer Linux-Tage seem not to have made differences between the free software and open source community while at the FOSDEM most visitors seem to consider being part of the free software community.

iii. While the visitors at the FOSDEM were mainly experienced hackers deeply rooted in the FOSS community the visitors at the Chemnitzer Linux-Tage were mainly students just starting being part of the FOSS community

A Goodness of fit test shows that the subjective assumption about community affiliation is supported by the data. Correspondingly, there is a highly significant difference between the Chemnitz and the Brussels population (see Figure 109).[603] In the Brussels sample the free software community is clearly bigger while the open source community is more important in the Chemnitz sample. This makes clear that concerning the community orientation the Brussels and the Chemnitz sample are different.

	Free Software community	Open Source community	I do not care	Missing / I do not know[a]	N
Brussels	45.57%	22.78%	24.05%	7.59%	79
Chemnitz	17.35%	41.84%	32.65%	8.16%	98
Brussels and Chemnitz	29.94%	33.33%	28.81%	7.91%	177
FLOSS study[b]	43.11%	29.31%	17.41%	10.17%	2774

a "I do not know" was no answer option in the FLOSS survey

Figure 109: Community affiliation and surveyed cluster

Concerning community affiliation there is no statistical significant difference between the population of the FLOSS study and the population of the Brussels sample.[604] This suggests that the FLOSS and the Brussels

[603] Chi-square test with 1 *df*, P = 0.005.
[604] Chi-square test with 1 *df*, P = 0.5.

population are comparable concerning their community affiliation. The Chemnitz population and the FLOSS population differ significantly.[605]

The Brussels and the Chemnitz population differ not only in their community affiliation but also in demographic characteristics. Subjects of the Brussels sample declared significantly higher English skills[606], higher age[607], higher education[608], higher gross income[609], and they were less frequently students[610] than the Chemnitz population. The experience in FOSS development is higher in the Brussels sample than in the Chemnitz sample. However, contrary to the subjective assumption the difference is not significant.

Response rate

The response rates for the two conferences differ sharply (see Figure 110 and footnote 612 for a methodological discussion about reported response rates). The response rate in Brussels was about 12% while it was about 43% in Chemnitz. The overall response rate is about 20%.

One reason for the different response rates is that in Brussels much more surveys were handed out that inhibited to convince prospective subjects personally to fill out the survey. In fact, because many developers approached the info desk simultaneously there was sometimes no time to talk with the prospective subjects. In Chemnitz the place where the survey was handed out was less crowded and therefore there was more time talking to a passerby. The opportunity in Chemnitz to fill out the survey directly at a table seems to be another major factor for the higher response rate. It can be assumed that many developers in Brussels who agreed to participate in the survey finally forgot to fill out the survey or just found more interesting tasks to do at the conference. It could also be speculated that subjects in Chemnitz had less experience with FOSS studies and were therefore probably more curious to participate. Additionally, the different response rates may be due to cultural reasons. In the international ISSP 2003 study the response rate in France was 17% while for the same study in Germany the response rate was 40% (ZUMA 2005: I-48 and I-50).[611]

	Conference participants	Distributed	Returned	Returned in %
Brussels	3500[a]	673	79	11.74
Chemnitz	2500[b]	227	98	43.17
Total	**6000**	**900**	**177**	**19.67**

a (Huber 2006: 90). The number of conference participants is estimated because there was free entry to the whole conference.

b (Sontag 2006)

Figure 110: Response rates

[605] Chi-square test with 1 *df*, P = 0.001.

[606] One-way ANOVA, P = 0.011 (Brussels average of 4.21, Chemnitz 3.84).

[607] One-way ANOVA, P = 0.063 (Brussels average of 30 years, Chemnitz 27 years).

[608] One-way ANOVA, P = 0.005 (Brussels average of 4.03, Chemnitz 3.44).

[609] One-way ANOVA, P = 0.012 (Brussels average of 2.69, Chemnitz 2.19).

[610] One-way ANOVA, P = 0.047 (Brussels average of 1.70, Chemnitz 1.55; 1 = study, 2 = no study).

[611] The methodological settings of the national ISSP surveys were different. This may also have contributed to the different response rates.

Whether the response rate of a survey is adequate is not easy to find out. It depends on the characteristics of the survey and how it is distributed (Deming 1960: 67; see also Schnell 1997: 29-43). Many mailed questionnaire surveys have response rates between 10% and 20% (Bailey 1994: 169; see also Spector 2001: 15). The two BCG studies had a response rate of 34.3% and 30.0% (Lakhani & Wolf 2005: 8) and a study of Apache HTTP server users had a response rate of 17% (Franke & von Hippel 2003: 1205). The study of Hars and Ou (2002: 31) had a response rate of 21%. Compared to other FOSS studies a response rate of 20% seems to be okay. All the more the number of eligible distributed surveys is computed in the most conservative way.[612] Particularly in Brussels the number of eligible distributed surveys is clearly too high.

In chapter 9.2 the optimal sample size for a 5 group ANOVA was computed to be 160.[613] Due to missing values and because of an equal distribution of subjects into groups cannot be assumed a target sample size of 200 subjects was determined. With 177 subjects participating in the survey the target sample size is not reached. However, the number of subjects seems to be large enough to conduct statistical analysis. While some of the other FOSS studies had less subjects (e.g. Hars & Ou 2002; Hertel, Niedner, & Herrmann 2003) other studies had more subjects (e.g. Ghosh 2005; Lakhani & Wolf 2005).

Reactions about conducting a paper survey

The decision to not conduct the survey online (see 9.4) was supported by the feedback given by subjects at both conferences. Several subjects stated that they had not participated in an online survey but that they will do so because the researcher took the time to appear in person. Additionally, many developers gave an interesting feedback on the subject that provided a good insight into the FOSS community and facilitated interpretation of the results.

10.2 Survey structure

In the following, the structure of the survey is presented (see also Appendix I). The survey consists, next to an introduction, of seven sections:

---

[612] The reported response rate is comparable to the RR2 definition of the American Association for Public Opinion Research (AAPOR 2006: 32). The RR2 is the standard for reporting response rates (for a definition of response rates see Schnell 1997: 20). The RR2 response rate is defined by the number of partially or completely returned surveys divided by the number of *eligible* distributed surveys (AAPOR 2006: 3). While the number of distributed and returned surveys is quite accurate it is not known how many of the distributed surveys are ineligible. A corresponding real example can be reported from the FOSDEM. Several empty and not yet distributed surveys got (uncounted) lost in a coffee cup accident. While they count as distributed they would be clearly ineligible. However, it is not exactly known how many surveys were lost and that is why all were counted as eligible. Another example is subjects who took the survey with them overnight but forgot to bring it back. These subjects asked for another survey and returned only one survey. Unfortunately, such cases were not counted accurately. In the RR2 definition such cases of unknown eligibility count for the response rate. An estimation of unknown eligibility as specified in the RR4 definition seems to be unnecessary because the increased accuracy is of low value.

[613] In fact, because the ANOVA was not run for five but for four groups the optimal sample size is 144 (see footnote 566).

Introduction

The introduction of the survey starts with explaining the goals of the study.[614] Subjects were then asked to refer their answers to the FOSS project he or she spent most time on during the last year. Thereby the following methodological problem was minimized: FOSS developers may be engaged in more than one project. The motivation of an individual may be different contingent on the project in question. In consequence, the answer may be correspondingly compromised.

The introduction includes also the time it takes for completing the survey. Observation at both conferences showed that the specified fifteen minutes were quite a suitable estimation. In the introduction there are also instructions how to return the survey. The person conducting the survey is shortly introduced and the possibility for posing questions is given by providing an email. Finally, there is a note including an URL that the results will be published under a GNU Free Documentation License / Creative Commons (attribution/share-alike) double license. This should show compliance to community rules. Additionally, it is assured that the data will solely be cited anonymized.

Section one: Community affiliation

Section one of the survey asks the subjects to state whether they consider being part of the free software community, the open source community, or whether they do not care. The reason for including this item was threefold. *Firstly*, the orientation towards one of these communities is thought to be a major variable indicating the Weltanschauung of the developers. The difference between the communities is rooted in a philosophical base (see chapter 2.1). *Secondly*, this item should make implicitly clear that the survey addresses both communities. There is anecdotal evidence that some subjects belonging to the free software community will not fill out the survey if only the open source community is addressed. *Thirdly*, because the wording of this item is almost similar to the one in the FLOSS study (Ghosh et al. 2002a), comparisons between these two studies will be possible.

Section two: Project characteristics

Section two of the survey is related to the characteristics of the FOSS project the subject contributes to. This includes the toughness of peer review, the satisfaction with project participation, interactional justice, the influence of companies, and received feedback. The project importance is queried in respect to the entire FOSS community, to the developers, and in respect to users. To research formal and informal decision making processes the survey asks how project positions are appointed. It is further asked to specify the project software, to indicate the number of programmers contributing, and the used development tools. In this section the phase of development is interrogated as well as how the project credits to contributors. It is finally asked what the project licenses are, whether subjects are happy with them, and if the software is included in the latest official release of at least one major Linux distribution.

[614] The comment that the data collected by the survey is needed for a Ph.D. thesis was added after the FOSDEM conference. In discussions with subjects at the FOSDEM it turned out that helping a Ph.D. student finishing his thesis was a major argument why subjects completed the survey (see also Schnell 1997: 253-254). At the same time it is not thought that this small adjustment lowers the comparability between the Chemnitz and Brussels samples.

Section three: Engagement

The items in section three are concerned with the engagement of the subject in the FOSS project. This includes the roles hold in the project, the starting year of contribution, the average hours of contribution per week, the average hours they are paid for, and a forecast whether they will in increase or decrease their contributions in the future.

Section four: Motivation

In section four of the survey subject's motivation is surveyed. There are each three items to explore fun seeking, investment in reputation, investment in human capital, peer affiliation, and software customization. In addition, there are four items about pro-social behavior.

Section five: Business interests

Section five refers to the relationship between project engagement and business interests. It is investigated whether there is such a relationship at all and if yes whether it is associated to the subjects own or the employee's business. If there is a business interest, it is asked to which extent project engagement can be attributed to functionalities, bug fixes, know-how to fork the project, and a good reputation in the FOSS community.

Section six: Basic need satisfaction

In section six of the survey basic need satisfaction such as autonomy, competence, and relatedness in the project is researched.[615] However, it is thought that labeling this section correspondingly would cause a major bias based on social desirability. Therefore, it was labeled slightly different without changing the meaning.

Section seven: Demographics

The last section of the survey is concerned with demographic information such as age, programming experience, working hours, education, the relationship to IT, English knowledge, and monthly gross income. At the very end of the survey there is the possibility to note down the email in order to get the survey results and to make some comments on motivation in FOSS projects.

10.3 Item description

After having discussed the structure of the survey, in the following the items finally included in the survey are presented (see also Appendix I). Certainly, the original item pool was considerably larger than the one actually used in the survey (see also DeVellis 2003: 66). To minimize the time needed to fill out the survey some concepts are not surveyed and the number of items per scale lowered. This may have compromised in some cases reliability and validity. Items of a scale should be comprehensive to cover the entire space of the latent

[615] The actual level of basic need satisfaction does not only depend on the social context provided by FOSS projects but also on characteristics of the individuals. However, including corresponding items is beyond the scope of this study.

variable and inclusive to fully capture the construct's domain of interest (Diamantopoulos & Siguaw 2006: 266-267).

10.3.1 Section one: Community affiliation

Item 1 surveys community affiliation. As already described there is within FOSS development a distinction between the free and the open source software community (see chapter 2.3.2). The item related to community orientation was adopted from the FLOSS study (Ghosh et al. 2002a: 50-59; 2002c, item 1). There are two small differences: Analogous to the wording in the whole survey the item refers to the term "FOSS" and not to the term "open source/free software". To comply with the structure and wording of the survey the term "the following questions" is substituted with the term "the questions below". There are no reasonable arguments why these two small differences should have any significant influence.

10.3.2 Section two: Project characteristics

Item 2 refers to the difficulty of an outsider getting code into the project. This item serves concurrently as a proxy for the toughness of the peer review process. The item refers to outsiders because for an insider knowing the community and familiar with the problems it may be easy to get code merged even it is difficult for anybody else. While designing the item special attention was giving to the wording so that it applies to projects using centralized version control[616] such as the Concurrent Versions System as well as to projects using decentralized version control[617] such as GNU arch or Git (see also Brown 2003: 16; Linux Weekly News 2005).

Item 3 serves as a proxy for satisfaction with past project participation. The item is based on an item of Clary et al. (1998: 1526) that measures satisfaction with a volunteer activity (see also Bruner, Hensel, & James 1992: 420).[618] In the process of survey development and after the pretests the wording was considerably changed.

Item 4 is an item to measure the extent to which the peer review process is based on a democratic voting process.

Item 5 measures the extent to which decisions are explained.

Items 6 and 7 are related to the involvement of companies in the project.

Item 8 measures the amount of positive feedback a subject gets in the project.

Items 9, 10 and 11 are measures of project success (see also item 20). It is argued that a project that would be missed if it were to disappear is successful. It may be that different groups in the FOSS community evaluate the success of a project differently. A developer may consider a buggy application successful because it

[616] In projects with a centralized version control developers with write access have to *push* the source code to shared servers.

[617] In projects with a decentralized version control each developer works with his own local repository of the source code. Other members of a project, such as for example Linus Torvalds in the Linux kernel, may *pull* source code from several different project members to their own repository. This obsoletes a central authority, the need for administering write permissions, and the problem of simultaneous file change. In consequence, there is per se no reference version of the source code but every new developer can freely join development without applying for writing access.

[618] The item of Clary et al. (Clary et al. 1998: 1526) is: "On the whole, the volunteer experience was very positive for me".

provides some state of the art features. Such features may be useless for a dummy user looking for stability and usability. It also may be that an audio player has no major bugs while providing only the usual feature. A developer may consider this project successful and miss if it were to disappear. However, the whole community probably does not even recognize the disappearance of the same project because there are alternative FOSS audio players. It is therefore necessary that for measuring project success it must be distinguished between a community, a developer, and a user perspective.[619]

Although it is very difficult to find FOSS success measures (Annabi, Crowston, & Howison 2003) that have no shortcomings (see Luethi 2006: 10-11) it is thought that the presented success measures here and in item 20 are the best available measures for success.

Item 12 is an item to measure governance mechanisms of a project in respect to the appointment of special positions. The need for special positions may be the result of horizontal or vertical division of labor (Frese 1995: 50; Kosiol 1962: 85-86). In discussion with FOSS community members three different mechanisms how members get a special position within the project could be identified: i) appointment by election, ii) hierarchical appointment by member holding a special position, and iii) appointment by informal consent. It is thought that many projects are too small for having a horizontal and especially a vertical division of labor. Therefore, there is also an answer option to indicate that the project is too small for special positions.

Item 13 specifies the software project. The corresponding items in the FLOSS study (Ghosh et al. 2002c: item 11 and 12) could not convince. In consequence, new categories had to be created.

Item 14 asks how many developers currently contribute source code actively. Because there is no formal membership in most FOSS projects subjects are asked to note down the *approximate* number of developers.

Item 15 is an item about development tools used in the project[620]. It is thought that the usage of some tools indicates certain project characteristics.

- The non-use of a versioning system may indicate a very small developer base because this tool seems to be necessary if several developers work together.[621]
- A bug tracking system seems to be an indicator for the user-base of a project. Above a certain number of users it seems to be necessary to have such a system.
- Automated regression is non-trivial to implement and indicates the level of knowledge represented in the project.

In the following, the different tools are shortly described.

Versioning system: Version control systems allow managing multiple revisions of the same unit of information (see Raymond 2004: 364-369). This tool is needed if several developers work together.

Bug tracking system: Bug tracking software manages software bugs. Users can enter and search bugs based on categories such as area of functionality, state of solution, priority, or severity. Some bug tracking software specifies an automated workflow. Without a proper bug tracking system bugs get lost or are poorly prioritized.

---

[619] Similarly, in their discussion of FOSS success measures Goldman and Gabriel (Goldman & Gabriel 2005: 101) implicitly assume that there are the three different types of stakeholders namely users, developers, and the community.

[620] For an overview over different development tools used in FOSS projects see Robbins (2005).

[621] However, it cannot be inferred that the usage of a versioning system indicates that many developers participate in the project. It may make sense to use a versioning system even if there is only one developer.

Automated regression testing: Regression testing intends to uncover bugs that occur if software functionality, that previously has worked, stops working in a successive release. One method to do so is rerunning tests whether previously fixed bugs remerges.[622] In software development it is considered good practice to run automated regression testing although its implementation is not trivial. Brooks (1995: 122) states:

> *As a consequence of the introduction of new bugs, program maintenance requires far more system testing per statement written than any other programming. Theoretically, after each fix one must run the entire batch of test cases previously run against the system, to ensure that it has not been damaged in an obscure way. In practice, such regression testing must indeed approximate this theoretical ideal … .*

In the study of Jørgensen (2000; Jorgensen 2001) 21% of FreeBSD developers run a regression test on the latest build they obtain.

Mailing lists: This mode of communication is based on email. It is used to distribute information to numerous (hopefully subscribed) recipients. Many mailing lists are open to everybody who is interested; some mailing lists require approval from the mailing list owner.

IRC: Internet Relay Chat is a text based type of instant communication over the Internet. However, like email or telnet it does not belong to the World Wide Web. Usually it is used for group communication.

Item 16 measures the development phase. At first glance, one may consider using the terminology "alpha", "beta", "release candidate", and "gold"[623] for measuring development phase. However, there are known software projects that are "perceptual beta". Such software is always termed as beta even it is already highly used for production. A prominent example is software produced by Google. It seems therefore that the terminology presented above works well within software industry but is not usable within the FOSS community. This is most probably related to the observation that (long-range) planning and formalities are not very common in FOSS development. To avoid these difficulties the development phases are defined as followed:

Planning: In the planning phase there is not much more than an idea for software development. Because it is easy to code some very preliminary features there will not be many projects in this category.

Proof of concept/for developers: In this phase of development it is shown that an idea is feasible. There must be in consequence some written source code. However, the source code may be buggy and does not need to be runnable.

Basic feature complete: Software that is runnable with the most elementary features implemented qualifies for this phase. However, because the set of features is yet constricted it is hardly suitable for regular end-user.

Ready for end-user: Software is ready for regular end-user if the software may be used in production mode and efficiently.

Feature complete/mature: If most reasonable features are implemented a software application may be considered mature. This is thought to be quite a rigorous description because there are often features that could be added. Therefore, it is not likely that many developers contribute to a project in this phase.

[622] There are three reasons why this may be fruitful: *Firstly*, a bug fix may get lost, *secondly* a bug fix is fragile and does not work anymore after a change has been made, and *thirdly* the fault that caused the original bug is made again.

[623] For a description of the term "gone gold" see footnote 534.

Item 17 identifies if and how contributors get credits. Means to credit contributors actively are mentioning on websites, mentioning in credit files, or write access to source repository (see also item 15 and footnotes 616 and 617).

Item 18 identifies the license model of a project. The source code of a FOSS project may be licensed under more than one license. An example is the web browser Mozilla Firefox that is released under a MPL/LGPL/GPL triple license.[624] To cope with such cases there is the possibility to note down up to three different licenses.[625]

Item 19 is an item to find out whether subjects are happy with the license model and if not which license they would prefer.

Item 20 is another measure of project success. It is assumed that software included in a major Linux distribution is more successful than one that is not (Luethi 2006: 11-13). The binary choice to include another piece of software in a Linux distribution such as Debian, Ubuntu, or SUSE is linked with costs because this includes assembling and testing. It may also be that a security problem or unobserved incompatibilities are incorporated. If the software is not Linux compatible an answer option provides the possibility to declare so.

10.3.3 Section 3: Engagement

Item 21 refers to the roles a subject holds in a project. Development of FOSS does not only include coding features but also fixing and reporting bugs, reviewing and commenting source code, creating graphics, writing FAQ's, translating, marketing and promoting, donating, helping users, and managing the project. In fact, a study on the activity of developers in commercial projects reveal that developers spend only about half of their time with coding (Perry, Staudenmayer, & Votta 1994: 40). While some of the roles in FOSS development are to some extent related to each other (e.g. writing and commenting source code) other roles are not (writing source code and creating graphics). Most probably a regular member of a FOSS project handles different task. In consequence, multiple answers are possible.

Item 22 asks in which year the subject started to contribute to the project (adapted from Ghosh et al. 2002c). Based on this item the project experience in years can be computed.

Item 23 measures how many hours a week a subject worked last year on the project on average (adapted from Ghosh et al. 2002c; Hars & Ou 2002).

Item 24 measures how many hours a week a subject was paid for project participation. Based on items 23 and 24 the percentage of paid hours can be computed. Thereby the question of payment is not reduced to the binary choice paid/not paid. It is therefore possible to distinguish between part- and full-time payment.

Item 25 is an item that measures the intention to decrease or increase the level of contribution in the future.

As can be seen in Appendix I item 25 is on the very top of page 2 of the survey. This caused a major problem insofar as the missing values are approximately two to four times higher than for items 2 to 11 which are thought to be similar difficult to answer. That the bad layout is responsible for the high number of missing

[624] See chapter 2.5 for the discussion of different licenses and chapter 2.5.3 for the discussion of the Mozilla case.

[625] There seems to be no major FOSS project with more than three licenses.

values is corroborated by the observation of subjects filling out surveys at the conference in Chemnitz. Many subjects simply overlooked the item.

10.3.4 Section four: Motivation

Each of the six different motivation types are measured by a scale consisting of three items. An exception is the pro-social behavior scale that consists of four items.[626] The items were extracted from different sources (Clary et al. 1998; Deci & Ryan 2006; Hars & Ou 2002; Roberts, Hann, & Slaughter 2006). Originally, it was intended to use standardized and inherently consistent items. However, pretesting showed that the corresponding items were hardly accepted by FOSS hackers. This supports the recommendation of Comrey (1988: 754) to not use standardized items because they are hardly "in the right form". In consequence, the standardized items were adapted to the FOSS topic.

Item 26 (Enjoy programming) is an item measuring fun seeking.

Item 27 (Code elegance) is an item measuring fun seeking.

Item 28 (Better job) is an item measuring reputation.

Item 29 (Better than course) is an item measuring human capital.

Item 30 (Innovation) is an item measuring pro-social behavior.

Item 31 (License force) is an item measuring pro-social behavior. The values of the item are recoded reversely.

Licenses such as the GNU GPL do actually not force to give source code back. *Firstly*, Section 2b of the GNU GPL license (FSF 2006c) refers to software that is distributed or published. Therefore, if software is not distributed or published the source code does not have to be revealed. *Secondly*, Section 3 of the GNU GPL requires that software must be accompanied by the source code or an offer to provide the source code. The GNU GPL does not require giving the source code back. The original author of the source code has to make an effort himself to get the modified source code. The GNU GPL does therefore *legally* not force giving source code back. However, the license terms have similar effects *in practice*. Therefore, it may be sound to ask the question although it is legally not demanded to give source code back.

Item 32 (Peer appraisal) is an item measuring peer recognition.

Item 33 (Fork) is an item measuring personal requirements.

Item 34 (Resulting code) is an item measuring fun seeking.

Item 35 (Bad code) is an item measuring peer recognition.

Item 36 (Talented programmers) is an item measuring reputation.

Item 37 (Personal needs) is an item measuring personal requirements.

Item 38 (Proprietary software) is an item measuring pro-social behavior. The values of the item are recoded reversely.

Item 39 (Judge skills) is an item measuring reputation.

[626] The reason that pro-social behavior is measured with four items is that it was originally planned to make additional analysis with this scale. However, do to time and place restrictions this analysis was omitted.

Item 40 (Programming skills) is an item measuring human capital.

Item 41 (Feedback importance) is an item measuring human capital.

Item 42 (Fix problems) is an item measuring personal requirements.

Item 43 (Concessions) is an item measuring peer recognition.

Item 44 (Ethical problem) is an item measuring pro-social behavior.

10.3.5 Section five: Business interests

Item 45 inquires whether there is a relationship between project participation and business interests or not. If there is such a relationship it is asked whether the subject is paid for project participation by an employer or whether the own business is related to the project.

Item 46 refers to the reason for project involvement of the employer or the own business. It asks how much project involvement is motivated by having certain functionalities implemented. This item is similar to Item 37 (personal requirements).

Item 47 refers to the reason for project involvement of the employer or the own business. It asks how much bug fixing is a reason for project involvement. This item is similar to Item 42 (personal requirements).

Item 48 refers to the reason for project involvement of the employer or the own business. It asks how much the ability to fork a project if it is necessary doing so is a reason for project involvement. This item is similar to Item 33 (personal requirements).

Item 49 refers to the reason for project involvement of the employer or the own business. It asks how much maintaining or building up a good reputation within the FOSS community is a reason for project involvement.

10.3.6 Section six: Basic need satisfaction

The following items on work situations respectively basic need satisfaction are adapted from Deci et al. (2006). They measure basic psychological needs such as autonomy, competence, and relatedness. Following Deci et al. (2006) one item per score is worded negatively and consequently recoded reversely.

In other research the number of items used for measuring the three basic needs includes three (Sheldon & Elliot 1999: 488), nine (La Guardia et al. 2000: 373; Sheldon et al. 2001: 328), fifteen (La Guardia et al. 2000: 371), twenty - one (Deci et al. 2001: 934; Ntoumanis 2005: 446), and twenty-three (Baard, Deci, & Ryan 2004: 2051) items. Here a nine item scale is used.

Item 50 (Influence patch merging) is an item measuring autonomy.

Item 51 (Pressure use tools) is an item measuring autonomy. The values of the item are recoded reversely.

Item 52 (Programmers praise me) is an item measuring competence.

Item 53 (Express ideas) is an item measuring autonomy.

Item 54 (Communication) is an item measuring relatedness. The values of the item are recoded reversely.

Item 55 (Sense of accomplishment) is an item measuring competence.

Item 56 (Programmers would miss me) is an item measuring relatedness.

Item 57 (Sympathy for programmers) is an item measuring relatedness.

Item 58 (Show capabilities) is an item measuring competence. The values of the item are recoded reversely.

10.3.7 Section seven: Demographics

Items 59 to 68 survey demographic information. They are partly adapted from the FLOSS study (Ghosh et al. 2002a, 2002c).

Item 59a asks to note down the year of birth. Based on this item the age of the subjects can be computed.

Item 59b asks to note down the year in which subjects started programming. Based on this item the programming experience can be computed.

Item 60 is an item measuring indirectly the spare time of a subject by asking how many hours a subject spends working and commuting to that job (adapted from Unger 1991: 84-87). To include not only working hours but also commuting is not without shortcomings. Nevertheless, to do so is thought to be more valid than measuring working hours alone.[627]

Item 61 is an item measuring the highest level of completed education.

This item caused some problems because educational systems are not completely comparable across different nation states. The UNESCO observes that for "…for any given country the interplay of cultural traditions, local customs, socio-economic conditions, at the very least, will have resulted in a concept of education in many ways unique to that country…" (UNESCO 1997: 9) and that "… it is clearly not possible to directly assess and compare the content of the educational programmes in an internationally consistent way" (UNESCO 1997: 16).[628] Additionally, the wording for educational attainment is for each country different. The meaning of the term "college" in the United States and the United Kingdom is for example not equal.[629]

The International Social Survey Programme (ISSP) handles the problem described above with three different strategies and with three different items related to education (ZUMA 2005: I-174-192). The *first* strategy is to ask for years of full time schooling including university. This has the major disadvantage that the answer to this item is not common knowledge. This either leads to a bias because subject guess the answer or the time needed to complete the survey increases because subjects have to compute the answer. Another problem is that in some countries a class level can or must be repeated while this is not possible in other countries. It may therefore be that more years of schooling are either result of a repetition or of higher education. A *second* strategy is to adapt items to the specific education system of each country.[630] The different educational system

[627] To compute spare time one could also add the hours spent on household chores or similar activities. However, doing so would make it necessary to distinguish between necessary and voluntary activities. Doing so would be too laborious for this study.

[628] The "Education Indicators" (UNESCO 2003) and first of all the "International Standard Classification of Education" (UNESCO 1997) measures of the UNESCO are suited to compare different educational levels internationally. These indicators are only usable for macro data analysis. However, the methodological discussion about these measures was helpful solving the problem of researching education levels across different nation states.

[629] The U.S. Current Population Survey roughly merges 17 categories of educational attainment (U.S. Census Bureau 2005: 8-17, item 18h) into "Less than High School", "High School Graduation", "Some College", and "Bachelor's degree or more" (U.S. Census Bureau 2004). In the Current Population Survey (U.S. Census Bureau 2004: 3) the categories are presented slightly different and the category "less than High School" has to be inferred.

[630] For the ISSP country specific surveys are created anyway.

could then be, based on the International Standard Classification of Education definitions (UNESCO 1997), matched by experts. This way of measuring education is probably the silver bullet. However, to do so for this study would eat up to many resources.[631] The *third* strategy is to describe a certain level of education by its relative position within the education system. A corresponding answer category in the ISSP is for example “lowest formal qualification” (ZUMA 2005: I-180).

For this study a mixed strategy was used to solve the problem of measuring education. It is thought that university degrees are more or less comparable across different countries. The naming of university degrees is also thought to be universally understandable by corresponding subjects. The meaning of the term bachelor degree, master degree, and doctoral degree (Ph.D.) are known at least in Europe.[632] A Ph.D. student from Switzerland knows most probably that he is a Ph.D. student even if this is not the official terminology. Therefore, these three levels of education were taken as categories for university education.

Education levels below university are less comparable and understandable. It is assumed that the meaning of High School and College are not properly understood by most non-Americans. A subject from Switzerland does probably also not know what an A-level is (this wording was for example used in the FLOSS study, see Ghosh et al. 2002c). The education levels below university are therefore measured by their relative position. The relative education levels used in this survey are “compulsory education”, “qualification above compulsory education but no entry requirement for universities”, and “entry requirement for universities”.[633]

The interview item from the U.S. Current Population Survey "What is the highest level of school ... has completed or the highest degree ... has received?” (U.S. Census Bureau 2005: 9-7) is adapted for this survey.

A problem in many surveys about education is that individuals do not always distinguish between the highest education *completed* and the highest education *started*. To decrease the extent of the related bias the word “completed” is highlighted in the survey.

Item 62 measures whether a subject studies right now. Item 61 surveys the highest *completed* level of education. A student preparing for a bachelor degree should then answer Item 61 with “Entry requirement for universities”. However, based on this answer it is not known whether a subject is studying or working after having completed the entry requirement for university. In the FLOSS study (Ghosh et al. 2002a: 14) 17% of developers are students. Being a student seems therefore to be a major variable in FLOSS development.

Item 63 is an item measuring the relationship to IT.

Item 64 specifies the relationship to IT.

Item 65 measures the level of English proficiency.

Item 66 measures the monthly gross income before taxes or insurances are taken away.[634]

It is thought that monthly gross income is the most sensitive item in the survey. To cope with this problem three strategies were employed. *Firstly*, the answer options are categorized. These categories are not very fine

[631] However, for studies conducted online this is no major effort.

[632] In the Bologna process that standardizes the European higher education area many countries changed their system and introduced the bachelor, masters, and doctoral degrees.

[633] The items measuring relative education in the ISSP (ZUMA 2005: I-180) are not adopted because they seem to foster social desirable answers.

[634] It is thought that whether taxes and insurance are subtracted or not from the monthly income is relevant.

graded which introduces a certain imprecision. This imprecision reassures subjects that they do not provide sensitive information. However, this has the disadvantage that the data level is artificially lowered from cardinal to ordinal and that thereby information is lost. *Secondly*, the answer option “I prefer not answering this question” is provided. To do so has advantages but also disadvantages. Forcing subjects answering sensitive items may introduce errors if they stop answering correctly. The problem is that not forcing people answering this item gives them an easy way out which produces a high level of missing values. *Thirdly*, the question about the monthly gross income is almost at the very end of the questionnaire. If a subject stops filling out the survey because the question is too private not much is lost. At that point the subject has already answered all relevant questions.

This study intends being comprehensible for FOSS developers from different countries although there will be most probably a Eurozone bias. This give rise to currency and exchange rate problems. This problem is handled by using the US$ currency for the answer categories. It is thought that virtually all FOSS developers know the approximate exchange rate of their local currency to the US$.[635] For convenience an approximate exchange rate of 0.8 euro for one dollar is given for FOSS developers from the Eurozone.[636]

Item 67 is an offer to get the results of the study.

Item 68 is an open item asking for comments on motivation in FOSS development.

Gender

The frequently asked item about gender is omitted in this survey. The reason for this decision should be briefly explained. In previous studies almost no female developers participated. The percentage of female developers in nine different FOSS studies is between 1.1% and 5% (see Figure 4 and chapter 2.4). The median percentage of these nine studies is 1.9%. It can therefore be assumed that in this study approximately 2% of subjects are female. Based on the target sample size of 200 subjects approximately four female subjects can be expected. The observation of subjects passing in completed surveys in Brussels and Chemnitz suggests that this assumption is quite accurate.

For statistical analysis the estimated four subjects have to be subdivided into different categories (e.g. development phases). As discussed in chapter 9.2 cell frequencies in chi-square tests should not fall below five. Meaningful statistics can therefore not be computed with the item gender. Additionally, no hypothesis is related to gender. Consequently, an item about gender is omitted to save space in the survey.

10.4 Measurement model specification

An important aspect of scale development is the decision whether a construct is reflexive or formative (see chapter 9.5). In the following, the related decisions are made for this study.

[635] Because most developers are thought to come from the Eurozone using the euro as a currency does seem to be appropriate at a first glance. However, it is assumed that developers outside the Eurozone (e.g. developers from the US) do not know the approximate exchange rate of their local currency to the euro.

[636] The two clusters Brussels and Chemnitz were surveyed between February 25, 2006 and March 5, 2006. During this time period one dollar was exchanged for 0.8437 - 0.8319 euros (ECB 2006). The approximate exchange rate of 0.8 euro for one dollar seems to be enough accurate. The categorized answers are not very accurate anyway.

10.4.1 Motivation

For the motivation scale there are a first and a second order constructs. The second order construct “motivation” is formed by the six different motivation types namely fun seeking, reputation, human capital, pro-social behavior, peer recognition, and personal requirements. The construct is clearly formative. The six different types of motivation are not necessarily correlated to each other. An individual may be highly motivated by the fun of programming but not at all by producing a signal to get a job.

In the first order constructs the latent variables are the six different ideal motivation types. The items of the constructs are the items presented in chapter 10.3 (see also Appendix I). Whether the constructs are formative or reflexive is not as easy to determine as for the second order construct. The decision whether models are reflective or formative is based on the typology of Jarvis et al. (Jarvis, MacKenzie, & Podsakoff 2003: 203) already discussed in chapter 9.5. Based on this typology the first order constructs are characterized as follows:

	Direction of causality	**Interchange-ability of items**	**Covariation among the items**	**Nomological net**	**Model classification**
Fun seeking	Construct ▸ Item	▸ ◂	▸ ▸	□	Reflective
Reputation	Construct ▸ Item	▸ ◂	▸ ▸	🗗	Reflective
Human capital	Construct ▸ Item	▸ ◂	▸ ▸	□	Reflective
Pro-social	Construct ▸ Item	▸ // ◂	▸ ▸	🗗	Formative
Peer recognition	Construct ▸ Item	▸ ◂	▸ ▸	□	Reflective
Personal requirements	Construct ◂ Item	▸ // ◂	▸ // ▸	🗗	Formative

Legend					
	◂ Items cause the construct	▸ // ◂ No Interchangeability	▸ // ▸ No covariation	🗗 Different antecedents and consequences	*Indication for a formative construct*
	▸ Items are an effect of the construct	▸ ◂ Interchangeability	▸ ▸ Covariation	□ Same antecedents and consequences	*Indication for a reflective construct*

Figure 111: Model specification: Decision support

The model classification is not for all types of motivation completely straightforward and should therefore be taken with precaution. Based on the direction of causality, the interchangeability of items, covariance among the items, and the nomological net of the items it is assumed that:

- ***Fun seeking:*** Fun seeking is clearly a reflective construct. A hacker for whom it is important that the source code is elegant (item 27) does most probably also state that the satisfaction of seeing resulting source code is important (item 34).
- ***Reputation:*** The classification of investment in reputation is reflective. Individuals who hope to get a better job (item 28) should also be concerned about credits that can be used to judge skills (item 39).
- ***Human capital:*** The classification of human capital is reflective. Somebody who states that he learns more from contributing to FOSS than from taking a programming course (item 29) should also state that his FOSS participation sharpened his programming skills (item 40).
- ***Pro-social:*** The classification of pro-social motivation is formative although the related decision is not obvious. The level of agreement to the statement that FOSS development is one of the big innovations in the 20^{th} century (item 30) is not necessarily related to the statement to use proprietary software (item 38, reversely coded)
- ***Peer recognition:*** The classification of peer recognition is reflective. Hackers for whom it is important that peers value their contributions (item 32) are likely ashamed if these peers thought that their source code was bad (item 35).
- ***Personal requirements:*** The classification of personal requirements is clearly formative. A developer may have a need to fix bugs (item 42) but not necessarily to fork a project (item 33).

The first order construct is a mixed formative-reflective model (see Figure 112). Fun seeking, investment in reputation, human capital, and peer recognition are thought to be reflexive. Personal requirements and pro-social motivation are thought to be formative.

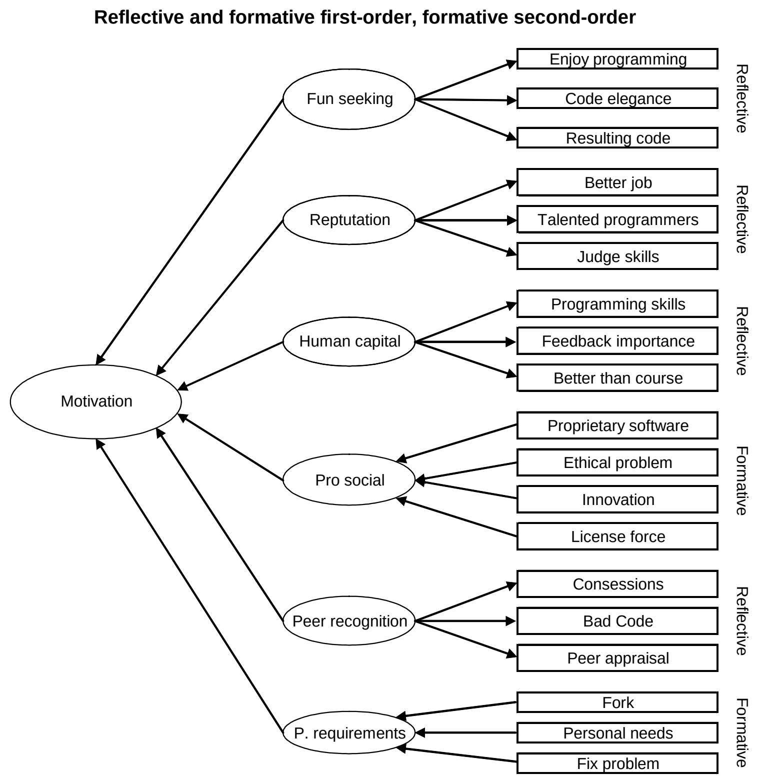


Figure 112: Model specification: Graphical representation

10.4.2 Basic need satisfaction

The model specification for the three basic needs autonomy, competence, and relatedness seems to be straightforward. Based on the direction of causality, the interchangeability of items, covariance among the items, and the nomological net of the construct items it is assumed that all three basic needs are measured by a reflective model.[637]

10.5 Data handling

In the following two aspects of data handling are discussed.

Coding

The coding of important items related to the main hypothesis such as phase development and motivation was proofread twice.

Some items are worded negatively. To enhance interpretability the corresponding items were recoded reversely.

[637] Implicitly, Baard et al. (Baard, Deci, & Ryan 2004: 2051) and Deci et al. (Deci et al. 2001: 934) also specified the three basic needs as a reflective model.

Contingency cleaning

Two parts of the survey have contingency items that should not be answered by all subjects. Whether a subject is supposed to answer an item is contingent on the answer to a previous item (Bailey 1994: 347):

- Only subjects who stated in item 45 that there is a relationship between business interest and project participation should answer items 46 to 49 that specify the business interest.
- Only subjects who stated in item 63 that their occupation or their studies has a relationship to IT should specify their IT relationship (item 64).

The instructions that accompany the two the contingency items ask to skip to the next item if there is no relationship to business interests respectively IT. However, such instructions are easily read over which makes contingency cleaning necessary (Bailey 1994: 347). Contingency cleaning involves checking the data to see whether only subjects have answered an item who are qualified to do so by means of an earlier response.

It can be recognized in Figure 113 that about half of the subjects who should not have specified the business relationship (Items 46-49) did so. The instruction to not answer items 46 to 49 if there is no business relationship failed clearly to work properly. This indicates a major layout error in the survey. A possible explanation why subjects answered these items without having a relationship to business interests is that they adapted the questions to their personal motivation to participate in FLOSS development. The reasons why a commercial firm participates in FLOSS development partly correspond to personal requirements and reputation investments of individuals.

Correct and wrong business relationship specification					
	Business relationship		**No business relationship / Missing relationship**		
	Question answered (correct)	Missing (wrong)	Question answered (wrong)	Missing (correct)	N
Functionalities	49	0	67	61	177
Bug fixing	49	0	65	63	177
Fork	48	1	65	63	177
Reputation	49	0	65	63	177

Figure 113: Items 46 to 49: Correct and wrong answer specification

The same problem as discussed above does not exist for the contingency item 64 (see Figure 114). There are only four subjects who specified their IT relationship although they indicated an item earlier that they have no IT relationship. Because the layout design is basically the same as for the business relationship the low misspecification must be attributed to the content of the item. If one has no relationship to IT it may be more difficult to make sense of answering the item.

Correct and wrong IT relationship specification					
	IT relationship		No IT relationship / Don't work or study / Missing relationship		
	Question answered (correct)	Missing (wrong)	Question answered (wrong)	Missing (correct)	N
Development	138	5	1	33	177
Support	138	5	0	34	177
System administration	138	5	1	33	177
Sales / marketing / constulting	138	5	0	34	177
Student	138	5	2	32	177
Professor / lecturer / assistant / teacher	138	5	0	34	177

Figure 114: Item 64: Correct and wrong answer specification

Although the wrong specification of the contingency items 46 to 49 and 64 is annoying it does not have any consequences for the results. Subjects who wrongly specified their relationship to business interests or IT are excluded from the corresponding analysis by recoding their input as missing.

11 Results: The reality test for the hypotheses

In the following, the results of the survey are presented. First of all, missing values are discussed. Afterwards, descriptive results are pinpointed. Finally, an analysis of the different scales precedes the presentation of inferential statistics used for hypothesis testing.

11.1 Missing values

The median of missing values of all items in the study is 7.3%. In the FLOSS study the median of missing values is 9.3% (Ghosh et al. 2002c). Compared to the FLOSS study the average fraction of missing values in this study, measured by the median, is therefore more advantageous.[638]

There are ten items with more than 10% missing values. Below it is speculated why the corresponding items have a higher proportion of missing values.

Item	**Missing (absolute)**	**Missing (in %)**	**Comments**
Bad code (35)	74	42%	The high proportion of missing values is due to a layout error in the printed version. The very last word “bad” was omitted in the surveys printed for the FOSDEM conference. After discovering the error a few surveys were corrected by hand. Subjects who answered the uncorrected version were coded as missing. As an exception, subjects who guessed the missing word correctly and filled it in the survey were not coded as missing. For the conference in Chemnitz the survey was reprinted without the error. Apart from the mentioned error it is thought that the item is a good one.

[638] The arithmetic mean of missing values in the FLOSS study (9.6%) is similar to the one in this study (9.5%). The difference between the two measures of central tendency is largely due to the two items with the highest percentage of missing values in this study.

Gross income (66)	54	31%	Items related to income are very sensitive. Therefore, the answer option "I prefer not answering" was provided. Corresponding answers were coded as missing and accounted for two-thirds of the missing values. The item in the FLOSS study concerning gross income had only 9% missing values (Ghosh et al. 2002c) that is comparable to the 11% missing values if the aforementioned answer option "I prefer not answering" is discarded. This questions the strategy to include a "no answer" option (see the discussion for Item 66 in chapter 10.3.7).
Project members (14)	32	18%	The item related to the number of project member seems to have been difficult to answer. Indeed, in most FOSS projects there is no formal membership. It is then not easy to ascertain whether lurkers actively following a project mailing list are project members or not. This explanation is backed up by the fact that many subjects answered the item by filling in questions marks or the answer "many" which is not interpretable and consequently coded as missing.
Contribution in the future (25)	30	17%	The item that asked whether the level of contribution increases or decreases in the future is speculative and partly not controlled by the subjects. To some extent the high level of missing values may be attributed to the difficulty of answering the item. However, observation of subjects in Chemnitz suggests that many overlooked the item because the corresponding layout is unfavorable.
English proficiency (65)	25	14%	The high level of missing values for the item related to English proficiency is rather surprising. It does not seem to be more difficult to answer than other items. An explanation may be that towards the end of the survey subjects either got tired answering items or lost interest. Answering demographic items is not interesting in general. This explanation is corroborated by the high number of missing values for demographic items in general. The items concerning education and the status of being a student have also a high percentage of missing values.
Hours working on project (23)	22	12%	An explanation for the high percentage of missing values concerning the average hours working on a FOSS project may be that the workload in a project may be highly unsteady. This makes it difficult calculating an average. The FLOSS study had almost the same fraction of missing values (11%) for this item (Ghosh et al. 2002c). This indicates that the reasons for the high level of missing values are not specific to this survey.

Project specification (13)	21	12%	The instructions for the item related to project specification request to select the most accurate answer. Many subjects have chosen two different specifications. Such answers were coded as missing because of interpretation problems. These cases accounted to a high degree for the high number of missing values. It cannot be determined whether this is due to a bad survey layout, to a bad answer categorization, or to item difficulty. The respective item 11 and 12 in the FLOSS study had with 10% and 14% a similar percentage of missing values (Ghosh et al. 2002c).
Spare time (60)	20	11%	It is assumed that there are two competing reasons for the high percentage of missing values for the spare time item. It may be the result of survey fatigue and little interest in demographic items (see discussion for item 65). A competing explanation is that subjects work very irregularly. For software developers, particularly for self-employed subjects, irregular working hours may be quite common. With irregular working hours it may be difficult to compute the correct answer.
Missed by community (9)	19	11%	The item whether the community would miss a project if it were to disappear is difficult to answer. It is a speculative item and refers to a heterogeneous group. This is thought to be the cause for the high percentage of missing values.

Figure 115: Discussion of high missing values

11.2 Results: Descriptive statistics

In the following chapter the results of the survey are presented by frequency tables and if appropriate by item analysis. Item analysis in psychometrics consists of item discrimination indices (only for items belonging to a scale) and item difficulties that will be briefly discussed first (Rost 2004: 97).

Item discrimination

A discrimination index is a measure of the effectiveness of an item to discriminate between high and low scores of a scale (Aiken 1985: 66). Item discrimination refers to the ability to separate subjects into different groups. For example in an examination a test question is ought to discriminate between high and low performers.

There are numerous methods expressing item discrimination (Aguinis, Henle, & Ostroff 2001: 31-32; Engelhart 1965; Oosterhof 1976: 145; Sim & Rasiah 2006: 70). Although some methods are more used than others no method is dominant. Here the corrected item-total correlation is used. This method is based on the Pearson product-moment correlation coefficient between the score of an item and the sum of the scores of the remaining items belonging to the same scale (Charoencholvanich & Pongcharoen 2005: 1196). The discrimination value may range between -1 and 1. Items with a positive discrimination value help to differentiate subjects. In contrast, an item with negative discrimination indicates that its non-use in the scale will help to distinguish subjects. A discrimination value near zero does not help splitting subjects into different groups.

It is usually assumed that the discrimination value of an item should be as high as possible (Ebel 1967: 128).[639] It is thought that item discrimination should be approximately at least 0.3 (Aiken 1985: 66).

Item difficulty

In psychological testing item difficulty is usually the percentage of examinees who select a correct answer (Aiken 1985: 461). Here and in many other tests there are no wrong or right answers. Item difficulty is then interpreted as the standardized arithmetic mean respectively the standardized level of agreement.[640]

Item difficulty may take value between zero and one. An item difficulty of zero represents in this study total disagreement to a statement while an item difficulty of one represents total agreement. The optimal item difficulty depends on the purpose of a study (Aiken 1985: 65). A test that intends to find subjects with an extreme agreement to a statement should have a low item difficulty so that almost all subjects disagree (adapted from Cronbach 1990: 211-212). The few subjects who have an extremely affirmative position will then be prominently visible. Similarly, if one intends to find the few subjects who heavily perform worse than the average the item difficulty should be very low. Analogously, if one intends to find only masterminds the item difficulty should be very high.

For the purpose of this study very high or low item difficulties are hardly usable because if everybody agree or disagree the item is probably trivial. Sim and Rasiah (2006: 69) state that extremely easy or difficult items have very low discriminative power. They therefore suggest using moderately easy or difficult items (Aguinis, Henle, & Ostroff 2001: 31; Sim & Rasiah 2006: 71). For this study items difficulty should approximately be between 0.2 and 0.8.

11.2.1 Community affiliation

It can be roughly said that one third of the subjects in the survey belong to the free software community, one third to the open source community, and one third does not care (see Figure 116). Looking at the two sample clusters Brussels and Chemnitz a major difference can be recognized. In Brussels approximately 50% belong to the free software community, 20% to the open source community, and 30% do not care or do not know the difference. In Chemnitz the affiliation to one of the two FLOSS communities is almost reversed. The open source community accounts for approximately 40% of the subjects and the free software community accounts for 20% of the subjects while the remaining 40% of the subjects do not care or do not know the difference. In short, it can be said that in Brussels the free software community and in Chemnitz the open source community dominated.[641]

[639] For a critical review of this assumption see Masters (1988).

[640] See also Crawford (1968: 104) for another computation method.

[641] The difference concerning the two communities is highly significant (Chi-square test with 1 df, P = 0.0001).

	Free Software community	Open Source community	I do not care	Missing / I do not know	N
Brussels	45.57%	22.78%	24.05%	7.59%	79
Chemnitz	17.35%	41.84%	32.65%	8.16%	98
Brussels and Chemnitz	***29.94%***	***33.33%***	***28.81%***	***7.91%***	***177***

Figure 116: FOSS community

Interestingly, concerning community affiliation the Brussels cluster is similar to the subjects who participated in the FLOSS study. The community affiliation in the Chemnitz cluster[642] and in the whole sample[643] are significantly different from the FLOSS study. This result points to two implications about sampling:

- *Firstly*, population characteristics found by this study should only be used for receiving an impression but not for corresponding hypothesis testing (see chapter 9.3). Sample selection, as shown by the difference between the Chemnitz and Brussels clusters, influences the results enormously. As discussed in the subchapter on sampling this does not inhibit hypotheses testing about causal relationships.
- *Secondly*, generalization of every other study on FOSS should be noted cautiously. Until now there seems to be no study without major sampling problems. The reasons for this are probably the special FOSS governance structures and mechanisms (see chapter 4.3). There is seldom a formal project membership that makes it very difficult to define the statistical population properly (see chapter 9.3).

Concerning community affiliation there seems to be a major difference between cultures (see also chapter 2.4). In the studies dominated by developers from Western societies (this study, FLOSS, FLOSS-US) the free software fraction is higher than in studies dominated by developers from Asian societies (FLOSS-Asia, FLOSS-JP). In Asian countries the open source community is considerably higher (see Table Figure 117). However, this difference may be due to sampling procedures.

	This study	FLOSS	FLOSS-US[b]	FLOSS-Asia	FLOSS-JP	WIDI
Free Software community	29.90%	43.10%	31.36%	25.60%	26.70%	41.10%
Open Source community	33.30%	29.30%	31.49%	42.10%	43.70%	52.07%
I do not care / I do not know[a]	36.80%	27.60%	37.14%	32.30%	29.60%	6.83%
N	177	2774	1540	133	547	5593

[a] The answer options in the FLOSS survey series and the WIDI survey referring to "no community affiliation" were slightly different.
[b] The FLOSS-US survey also includes developers from outside the US. Developers from Western Europe or Russia (61%) dominate the sample but less than in the orignial FLOSS survey (71%).

Figure 117: Community affiliation in different studies[644]

[642] Chi-square test with 1 df, P = 0.001.
[643] Chi-square test with 1 df, P = 0.01.
[644] David et al. (2003b: 2), Mitsubishi Research Institute (2004c; 2004d), and Robles et al. (2001a: 41).

11.2.2 Project characteristics

Easiness to get code into the project

About 16% of the subjects state that it is easy to get code into the project while 3% answered that it is very difficult (see Figure 118). The item difficulty of 0.58 indicates that it is on the average neither especially difficult nor easy to get code into the project.

Good decision to participate

The item difficulty of 0.85 indicates that looking back subjects consider their project participation as a good decision (see Figure 118). Virtually nobody regrets (2.26%) or strongly regrets (0.56%) this decision.

Voting procedures and code review

There is no clear tendency concerning voting procedures in the code review process (see Figure 118). For some projects there is strong agreement (7.34%) that there is some sort of voting procedure while in other projects this decision making process is absent (16.95%).

Project characteristics	Frequency tables						Item analysis
	strongly disagree				strongly agree	Missing	Item difficulty[a]
Easy get code into project	3.39%	21.47%	28.81%	25.99%	16.38%	3.95%	0.58
Good decistion	0.56%	2.26%	10.17%	27.12%	55.37%	4.52%	0.85
Voting in code review	16.95%	24.29%	28.25%	16.95%	7.34%	6.21%	0.49
Decisions explained	2.82%	15.82%	32.20%	28.25%	11.86%	9.04%	0.58
Company contribution beginning	44.63%	15.25%	12.99%	10.73%	10.17%	6.21%	0.30
Company contribution right now	30.51%	10.17%	22.60%	22.03%	7.91%	6.78%	0.41
Interesting comments	3.39%	19.77%	23.73%	33.90%	11.86%	7.34%	0.58
Missed by community	12.99%	24.29%	17.51%	15.25%	19.21%	10.73%	0.51
Missed by developers	4.52%	7.91%	14.69%	28.25%	36.16%	8.47%	0.72
Missed by users	2.26%	2.82%	12.99%	28.81%	45.20%	7.91%	0.80

N=177
[a] Standardized arithmetic mean ranging from 0 (total disagreement) to 1 (total agreement)

Figure 118: Item analysis of project characteristics

Explanation of controversial decisions

There are almost no projects (2.82%) in which controversial decisions are not explained (see Figure 118). The item difficulty (0.58) points out that there is a slight tendency towards explaining controversial decisions.

Company contributions

Companies hardly make important contributions at the project start (see Figure 118). For almost half of the FOSS projects there are no important contributions at the beginning at all. The item difficulty of 0.30 supports this observation. The item difficulty for recent company contribution is larger (0.41) which denotes that company contributions increase with ongoing project maturing. Commercial firms seem to be more interested investing scare resources in mature project and projects that keeps up than in new projects. This supports hypothesis 27 that the development phase is positively related to the involvement of commercial firms.

The above reported difference between “recent” contribution and contributions “at the beginning” should not exist for young projects. In these cases “recently” corresponds temporarily to “at the beginning”. This allows making a reliability and validity check. Although there is no item directly related to project age it can be measured indirectly. The project experience of developers is partly related to project age. A long project experience indicates clearly a high project age. By contrast a short project experience does not indicate a young project age. A short project experience may be the result of a young project age or it may be the result of a developer who just joined an old project. However, taken together it can be assumed that on the average developers with a long project history contribute to older projects than developers with a young project experience do.

As can be recognized in Figure 119 the difference between company contributions right now and at the project beginning increases with higher project experience of the subject. There is a positive and significant correlation between project experience and recent company contribution.[645]

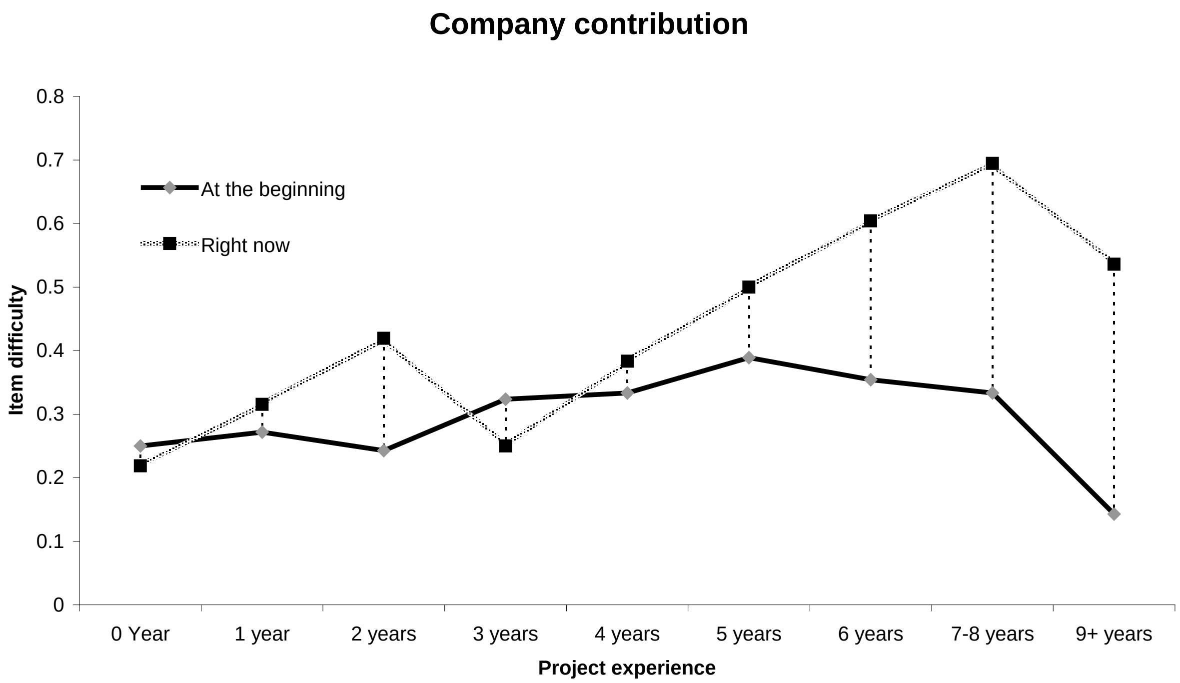


Figure 119: Project experience and contributions of companies

[645] Kendall-Tau-b correlation = 0.22, two-sided, P = 0.01, N = 157. An alternative explanation for this correlation is that more experienced project members have a better knowledge about companies’ project participation and report more likely the correct higher value.

Positive feedback

There are not many projects (3.39%) in which developers do not get interesting comments about their submitted code (see Figure 118). The item difficulty (0.58) indicates that there is a slight tendency that developers get interesting comments.

Missed if project were to disappear

Three similar items asked whether a project would be missed by the community, by the developers, and by the users if it were to disappear. These items measure indirectly project success. It is argued that a project that would be missed if it were to disappear is successful.

The response pattern is quite realistic (see Figure 118). The item difficulty for the item "missed by community" is with 0.51 considerably lower than for "missed by developers" (0.72), and "missed by users" (0.80). The subjects acknowledge that the disappearance of many projects would cause problems for the users and dolefulness for developers but that in many cases it would be hardly recognized within the whole FOSS community. Many members of high rated projects such as Debian[646] participated in the survey. It can therefore be assumed that the 20% of subjects who strongly agree that the project would be missed by the whole FOSS community may be realistic.

In the FLOSS-US study an item asked how known their current/most recent project is to the FOSS community at large. This item can be assumed as a rough proxy for project success. Certainly, a famous project is not necessarily successful. Software applications such as Duke Nuke Forever can be renowned for poor development performance. However, such examples are probably exceptions. Famous FOSS projects such as Linux, the Debian distribution, or the Apache HTTP server are successful and would be most probably also missed by the community. In the FLOSS-US study roughly 35% of developers answered that their project is famous or well known while 65% answered that their project is slightly known or unknown (David, Waterman, & Arora 2003b: 22). In this study 39% of subjects agree or strongly agree that the FOSS community would miss their project if it were to disappear while 42% disagree or strongly disagree. In consequence, it can be concluded that concerning project success the FLOSS-US study and this study are roughly similar.

Project positions

As anticipated in chapter 10.3 there is a high proportion (36.72%) of projects that are too small for having special positions (see Figure 120). In 24% of the remaining projects there are elections, in 34% special positions are appointed by a leader, and in 43% there is an informal consent.

[646] Some surveys were distributed and collected directly at the Debian booth (see chapter 10.1).

Governance mechanisms					
Elected	**Appointed**	**Informal consent**	**Too small**	**Missing**	**N**
12.99%	18.64%	23.16%	36.72%	8.47%	177
23.71%	34.02%	42.27%	-	-	97

Figure 120: Appointment for special positions

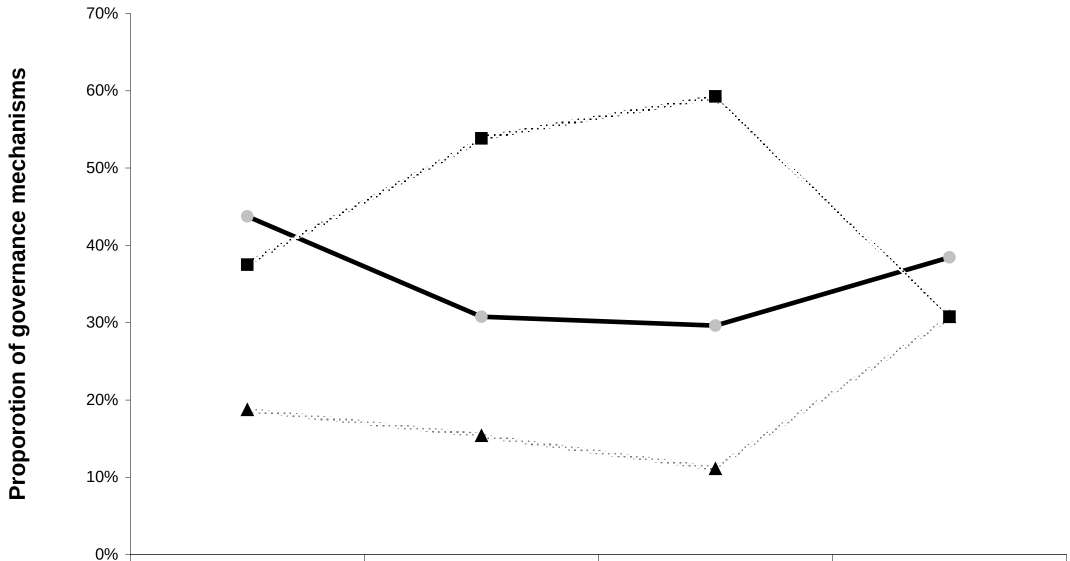
In Figure 121 it can be recognized that the valid percentage of subjects who answer that their project is too small for having special position decreases significantly from 72% for very small projects (1-4 developers) to 48% for small projects (5-9 developers)[647] to 16% for medium projects (10-25 developers)[648] to zero percents for big projects (>25 developers)[649].

Governance mechanisms	***Number of programmers*** *Very small project 1-4*			*Small project 5-9*			*Medium project 10-25*			*Big project >25*		
Elected	5.1%	5.2%	18.8%	7.7%	8.0%	15.4%	9.4%	9.4%	11.1%	28.6%	30.8%	30.8%
Appointed	11.9%	12.1%	43.8%	15.4%	16.0%	30.8%	25.0%	25.0%	29.6%	35.7%	38.5%	38.5%
Informal consent	10.2%	10.3%	37.5%	26.9%	28.0%	53.8%	50.0%	50.0%	59.3%	28.6%	30.8%	30.8%
Too small	71.2%	72.4%	-	46.2%	48.0%	-	15.6%	15.6%	-	0.0%	0.0%	-
Missing	1.7%	-	-	3.8%	-	-	0.0%	-	-	7.1%	-	-
N	59	58	16	26	25	13	32	32	27	28	26	26

Figure 121: Governance mechanism and number of programmers

The dominant governance mechanisms in very small projects are informal consent and appointment (see Figure 122).[650] In small[651] and medium[652] projects solely informal consent is the dominant governance mechanism. For big projects with more than twenty-five programmers the three governance mechanisms election, appointment, and informal consent are approximately equally important. Interestingly, elections are only important in big projects.

[647] Chi-square test with 1 df, P = 0.001.
[648] Chi-square test with 1 df, P = 0.001.
[649] No significance test possible because one cell frequency is zero.
[650] However, the difference between election and appointment respectively between election and informal consent is not significant.
[651] Informal consent is not significantly different.
[652] Chi-square test with 2 df, P = 0.008.



	1-4	5-9	10-25	>25
Appointed	43.75%	30.77%	29.63%	38.46%
Informal consent	37.50%	53.85%	59.26%	30.77%
Elected	18.75%	15.38%	11.11%	30.77%

Figure 122: Chart of governance mechanisms and number of programmers

Software specification

Subjects of the survey contribute most frequently to office applications/productivity tools and development tools/libraries (see Figure 123). Less frequently they contribute to server software, graphics/audio/multimedia, and kernel/driver. Not very astonishingly games are the least common type of FOSS projects subjects contribute to.

Software specification	Frequency
Office applications/Productivity tools	22.03%
Development tools/Libraries	19.21%
Server software	16.38%
Graphics/Audio/Mulitmedia	9.04%
Kernel/Driver	7.34%
Distribution	6.21%
Other	6.21%
Games	1.69%
Missing	11.86%
N	177

Figure 123: Software specification

Number of programmers

The median number of programmers is 5 that means that 50% of projects are smaller and 50% bigger.[653] Only 30% of the projects have more than 10 and only 17% of the projects have more than 25 programmers (see also Figure 124).

Percentil	Number of programmers
20	2
40	4
60	10
80	25
90	100

Figure 124: Number of programmers

In the FLOSS-US study the median number of programmers is 6 and 38% of the projects had more than 10 developers (David, Waterman, & Arora 2003b: 30). Concerning the number of programmers the FLOSS-US and this study are therefore similar.

The median number of programmers varies significantly contingent on the development phase (see Figure 125).[654] In the planning/proof of concept phase the median is 2.5 programmers, in the basic feature complete phase the median is 4, in the ready for end-users phase the median is 8, and in the mature phase the median is 10 (see also the number of developers in mature SourceForge project; chapter 2.4).

[653] The median is taken because there are extreme outliers. It is also assumed that it is harder to compute the number of programmers in bigger than in smaller FOSS projects. While one can easily decide whether 2 or 1000 programmers contribute to a project it is hard to decide whether there are 1000 or 2000 project members. In both cases the error may be 1000 programmers. While the arithmetic mean is very sensitive to such problems the median is not. The median divides the higher and lower half of a sample. If a subject wrongly indicates that the project has 2000 instead of 1000 programmers this has no effect on the corresponding median in this study.

[654] Kruskal-Wallis one-way analysis of variance, df $=3$, $P=0.0001$.

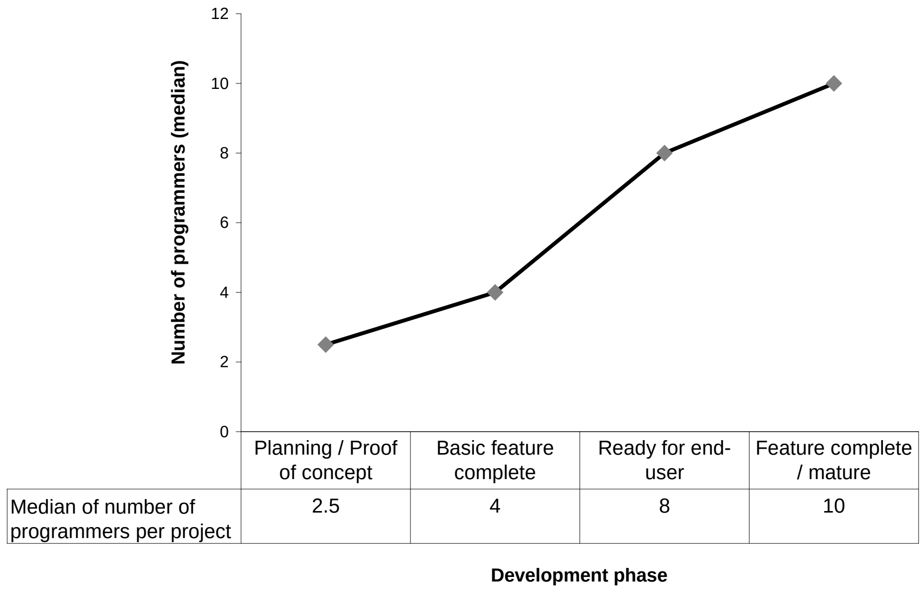


Figure 125: Chart of median number of programmers contingent on the development phase

Project tools

More than four-fifth of the subjects state that their project has a versioning system. Slightly fewer projects have a mailing list. Having a bug tracking system is also common and adopted by two-thirds of the projects. Automated regression testing and IRC chat is clearly less frequent and used in one-fourth of the projects. Project tools pooled in the category "other" are inter alia wikis and Internet forums.

Project tools	Frequency
Versioning system	85.45%
Bug tracking	66.06%
Mailing list(s)	77.58%
Automated Regression testing	26.67%
IRC	27.27%
Other	16.57%

Figure 126: Project tools

There are virtually no projects that do not adopt at least one project tool (see Figure 127). Projects use most frequently four different tools (mode). This indicates together with the median (3 project tools) that FOSS projects use a wide mix of tools for development.

Project tool count	Frequency	Cumulative Frequency
0 tool	1.13%	1.13%
1 tool	17.51%	18.64%
2 tools	19.21%	37.85%
3 tools	23.16%	61.02%
4 tools	25.99%	87.01%
5 tools	7.34%	94.35%
6 tools	2.26%	96.61%
Missing	3.39%	100.00%
N	177	177

Figure 127: Project tool count

Development phases

As can be seen in Figure 128 the phases “basic feature complete”, “ready for end-user”, and “feature complete/mature” consist roughly of about one-fourth of the subjects. There are considerably less subjects in the phases “planning” and “proof of concept”.

The above presented distribution of subjects in the different phases suggests to collapse the “planning” and the “proof of concept” phase into one phase. Thereby subjects are approximately equally divided in all phases.[655] To do so has two methodological advantages. *Firstly*, it is difficult to conduct empirical tests if there are groups with only few subjects (see also chapter 9.2). *Secondly*, if groups have an equal N power is maximized (Baroudi & Orlikowski 1989: 101).[656] Because of this reasoning the “planning” and the “proof of concept” phase are collapsed into one phase. In consequence, roughly one-fourth of subjects are in each of the four phases.

<table>
<tr><th colspan="7">Development phases in percents</th></tr>
<tr><th>Planning</th><th>Proof of concept / for developers</th><th>Basic feature complete</th><th>Ready for end-user</th><th>Feature complete / mature</th><th>Missing</th><th>N</th></tr>
<tr><td>5.65%</td><td>10.17%</td><td>21.47%</td><td>26.55%</td><td>26.55%</td><td>9.60%</td><td>177</td></tr>
<tr><td colspan="2">17.50%</td><td>23.75%</td><td>29.38%</td><td>29.38%</td><td>-</td><td>160</td></tr>
</table>

Figure 128: Development phases

[655] In an empirical study of Bales and Strodtbeck (1960: 629) on phases of a group the authors state that they have no basis for predicting the length of these group phases. In consequence, they assume that the phases have the same length.

[656] Baroudi and Orlikowski (1989: 101) report the example of a study with 108 subjects that are divided into two groups of 22 and 86 subjects. The power of a statistical test with this unequal distribution of subjects is the same as for a study with 70 subjects equally divided in two groups. In consequence, in the former situation 38 cases are “wasted”.

Credits

The most frequent mean for giving credits are websites (66%) followed by credit files (58%). Only about one-third of the projects credit the contributors by providing write access to the source repository. This may be partly explained by the fact that giving write access to the source repository makes only sense for centralized version control systems. The cases pooled in the category “other” include expressing gratitude personally by email, mentioning in the documentation, mentioning in mailing lists, and mentioning in the “about dialog”.[657]

Credits	Frequency	
Website	66.10%	68.02%
Credit file	57.63%	59.30%
Write access source repository	30.51%	31.40%
Other	6.21%	6.40%
Missing	2.82%	-
N	177	172

Figure 129: Modes to credit

Most projects (81%) have one or two different modes for giving credits (see Figure 130). Almost no project (4%) omits giving credits in some way. More than two different modes of giving credits are not very common (15%).[658]

Different modes to credit	Frequency	Cumulative Frequency
0 modes	3.95%	4.07%
1 mode	40.68%	45.93%
2 modes	38.42%	85.47%
3 modes	13.56%	99.42%
4 modes	0.56%	100.00%
Missing	2.82%	-
N	177	172

Figure 130: Frequency of different modes to credit

[657] The “about dialog“ is a dialog box in software that displays different information such as copyright statements and registration details or gives information about the developers.

[658] The count of different modes for giving credit should be taken with a pinch of salt because the list of possible modes may not be exhaustive.

Licenses

From the projects that have already made a decision concerning the license scheme 89% are licensed under a copyleft and the remaining 11% under a non-copyleft license (see Figure 131). From the copyleft licenses the vast majority were put under a strong copyleft license that is in this study exclusively the GNU GPL. The few FOSS projects under a weak copyleft license are licensed under either the GNU LGPL or the Mozilla Public License.

<table>
<tr><th colspan="5">Classified FOSS licenses</th></tr>
<tr><td>Strongly copylefted</td><td>72.33%</td><td>80.99%</td><td rowspan="2">89.44%</td><td rowspan="3">89.31%</td></tr>
<tr><td>Weakly coplefted</td><td>7.55%</td><td>8.45%</td></tr>
<tr><td>Not copylefted</td><td>9.43%</td><td>10.56%</td><td>10.56%</td></tr>
<tr><td>Not decided</td><td>10.69%</td><td>-</td><td>-</td><td>10.69%</td></tr>
<tr><td>N</td><td>159</td><td>142</td><td>142</td><td>159</td></tr>
</table>

Figure 131: Classified FOSS license

The difference between the distribution of licenses at SourceForge in 2006[659] and in this study is clearly not significant if it is only distinguished between copyleft and non-copyleft licenses.[660] However, if strong copyleft and non-strong copyleft licenses are compared the difference gets significant.[661]

Not astonishingly, the fraction of FOSS projects that has not made yet a decision about licenses is higher for projects not included in a major Linux distribution than projects included (see Figure 132).

<table>
<tr><th colspan="9">Classified FOSS licenses and inclusion in major Linux distributions</th></tr>
<tr><th></th><th>Included</th><th>Not included</th><th>Included</th><th>Not included</th><th>Included</th><th>Not included</th><th>Included</th><th>Not included</th></tr>
<tr><td>Strongly copylefted</td><td>76.56%</td><td>72.60%</td><td>80.33%</td><td>84.13%</td><td rowspan="2">91.80%</td><td rowspan="2">92.06%</td><td rowspan="3">95.31%</td><td rowspan="3">86.30%</td></tr>
<tr><td>Weakly coplefted</td><td>10.94%</td><td>6.85%</td><td>11.48%</td><td>7.94%</td></tr>
<tr><td>Not copylefted</td><td>7.81%</td><td>6.85%</td><td>8.20%</td><td>7.94%</td><td>8.20%</td><td>7.94%</td></tr>
<tr><td>Not decided</td><td>4.69%</td><td>13.70%</td><td>-</td><td>-</td><td>-</td><td>-</td><td>4.69%</td><td>13.70%</td></tr>
<tr><td>N</td><td>64</td><td>73</td><td>61</td><td>63</td><td>61</td><td>63</td><td>64</td><td>73</td></tr>
</table>

Figure 132: Licenses and inclusion in major FOSS distributions

Figure 133 shows that subjects who can decide whether they are happy or not with a license scheme are almost unanimously happy.

[659] See Figure 27.
[660] Chi-square test with 1 df, P = 0.25.
[661] Chi-square test with 1 df, P = 0.025.

<table>
<tr><th colspan="4">Happiness with FOSS licenses</th></tr>
<tr><td>Happy with licenses</td><td>84.66%</td><td>97.18%</td><td rowspan="3">87.12%</td></tr>
<tr><td>Not happy: Prefer non-copylefted instead of copylefted license</td><td>1.84%</td><td rowspan="2">2.82%</td></tr>
<tr><td>Not happy: Prefer non-copylefted instead of other non-copylefted license</td><td>0.61%</td></tr>
<tr><td>Can not decide</td><td>12.88%</td><td>-</td><td>12.88%</td></tr>
<tr><td>N</td><td>163</td><td>142</td><td>163</td></tr>
</table>

Figure 133: Happiness with license

Inclusion in major Linux distribution

Roughly speaking one-half of the projects are included in a major Linux distribution while the other half is not (see Figure 134). For about one-tenth the status of inclusion is not known or the application is not Linux compatible.

Inclusion in a major Linux distribution	Frequency	Frequency	Frequency
Included in distribution	40.68%	42.60%	48.00%
Not included in distribution	44.07%	46.15%	52.00%
Inclusion status not known / not Linux compatible	10.73%	11.24%	-
Missing	4.52%	-	-
N	177	169	150

Figure 134: Inclusion in a Linux distribution

11.2.3 Engagement

More than two-thirds of the subjects included in the study are involved in programming such as developing features (73%)[662] respectively fixing bugs (67%, see Figure 135 and Figure 136). A majority of 57% of the subjects both develops features and fixes bugs. About 15% only develop features, 10% only fix bugs, and a considerable fraction of 17% neither develops features nor fixes bugs. This implies that about 83% of the subjects who develop features also fix bugs and that roughly 85% oft the subjects who fix bugs develop features as well.

That a substantial fraction of subject neither develops feature nor fixes bugs is rather surprising. It makes clear that a substantial fraction of subjects is not involved in some kind of programming.

[662] In the FLOSS-US study 84% of the subjects were engaged in coding which is comparable to this study (David, Waterman, & Arora 2003b: 34).

		Feature developing: No	Feature developing: Yes	
Bug fixing	No	17.16%	15.38%	**32.54%**
	Yes	10.06%	57.40%	**67.46%**
		27.22%	**72.78%**	**100%** (N=169)

Figure 135: Engagement: Feature developing and bug fixing

More than half of the subjects (53%) help users that is not anticipated to this extent (see Figure 136). That almost half of the subjects (47%) review and comment source code shows the prominence of the peer review process in FOSS development.

Roughly 92% of subjects in the study have more than one role in their projects (see Figure 137). This is not surprising because many projects are small that makes it necessary that individuals take on different roles. Most subjects take on four different roles (mode).

Project roles	Frequency
Adding or developing features	72.78%
Bug fixing	67.46%
Bug reporting	54.44%
Helping users	53.25%
Reviewing and commenting code	46.75%
Writing FAQ	34.91%
Project management	34.91%
Marketing/Promoting	28.40%
Translating	25.44%
Creating graphics	17.16%
Donating	11.83%
Other	5.33%
N	169

Figure 136: Project roles

Number of project roles	Frequency	Cumulative Frequency
1 role	7.91%	8.28%
2 roles	10.17%	18.93%
3 roles	14.12%	33.73%
4 roles	18.64%	53.25%
5 roles	15.82%	69.82%
6 roles	11.86%	82.25%
7 roles	7.91%	90.53%
8 roles	3.95%	94.67%
9 roles	2.26%	97.04%
10+ roles	2.82%	100.00%
Missing	4.52%	-
N	177	169

Figure 137: Number of project roles

A principal component analysis with varimax rotation[663] of the 11 project roles (without "other")[664] revealed three different groups of closely inter-related project roles (see Figure 138). These three groups of projects roles with an eigenvalue above 1 are ***"coding"*** (adding or developing features, bug fixing, reviewing and commenting code), ***„promoting and supporting"*** (bug reporting, writing FAQ, translating, marketing and promoting, donating, helping users), and ***"managing and creating graphics"***. The three groups account for 48% of the variation.

In the FLOSSPOLS study a principal component analysis of 22 different activities in FOSS development revealed 6 distinct groups of project roles (Ghosh & Rüdiger 2006: 18-19). These groups of activities are organizing (similar to project management), bug fixing (similar to coding), politics (similar to marketing and promoting), translating, communication (similar to helping users), and supporting (similar to creating graphics and to write FAQ).

[663] Varimax rotation minimizes the number of items that have high loadings on each factor that in turn produces "'nice' results from the viewpoint of scientific interpretability" (Kaiser 1979: 33). Diamantopoulos and Siguaw (2006: 269) use varimax rotation to investigate the dimensionality of scales (see also chapter 11.3.1). Similarly, this rotation method is also used here for investigating dimensionality.

[664] If „other" is included four factors are extracted. The result remains similar. The only notable difference is that "translating" and "other" have their highest factor loading on the fourth factor.

Principal component analysis with varimax rotation			
	Factor loadings[ab]		
	Coding	***Promoting and Supporting***	***Managing and creating graphics***
Adding or developing features	**0.712**	-0.234	0.045
Bug fixing	**0.821**	0.022	0.016
Reviewing and commenting code	**0.482**	0.23	0.375
Bug reporting	0.391	**0.519**	-0.51
Writing FAQ	0.091	**0.663**	0.079
Translating	-0.081	**0.39**	-0.009
Marketing/Promoting	-0.002	**0.674**	0.125
Donating	-0.251	**0.587**	-0.037
Helping users	0.226	**0.509**	0.152
Creating graphics	-0.001	-0.003	**0.727**
Project management	0.239	0.255	**0.642**

[a] Rotated in 6 iteration; [b] Highest factor loadings are bold

Figure 138: Factor analysis of project roles

Project experience

The project experience of the subjects ranges between less than one year and fifteen years.[665] Almost two-thirds of the subjects in the study have a project experience between 1 and 3 years. A considerably minority (29%) has a project experience between 4 and 9 years. Only 4% of the subjects have a longer project experience than ten years. On the average project experience is 2 (median) respectively 3.1 (mean) years.

In the FLOSS-US study the average FOSS experience is 4 years (median, David, Waterman, & Arora 2003b: 16). It can therefore be stated that the subjects have a somewhat lower project experience than the ones in the FLOSS-US study.[666]

[665] Item 22 asks in which year subjects started to contribute. This was recoded into years of project experience to ease statistical analysis. Strictly speaking this recoding is imprecise. A subject who started contributing to a project in December 2005 would have responded to item 22 at the Chemnitzer Linuxtage in March 2006 with the answer "2005". This would have been recoded into one year of project experience while the true project experience is only three months. However, it is not thought that this imprecision has a major influence. Additionally, a finer graded item that includes months is too difficult to answer.

[666] The corresponding results are not completely comparable because this study queried project experience in respect to a certain FOSS project while the FLOSS-US study queried experience in respect to FOSS development in general.

Years of project experience	Frequency	Frequency	Cumulative Frequency
> 1 year	5.08%	5.49%	5.49%
1	27.68%	29.88%	35.37%
2-3	29.38%	31.71%	67.07%
4-5	14.69%	15.85%	82.93%
6-9	12.43%	13.41%	96.34%
10-15	3.39%	3.66%	100.00%
Missing	7.34%	-	-
N	177	164	164

Figure 139: Project experience

The distribution of project experiences shows that the phenomenon of FOSS development is relatively new. Although sharing and revealing of source code is older (e.g. the first BSD distribution was shipped in 1977, see chapter 2.2) only in the nineties the community gained momentum with projects such as Linux in 1991 and Apache in 1995. The spreading of cheap Internet connections and low total costs of ownership of computers are probably responsible for this observation.

Working hours for project

On average the subjects in the study work five hours (median) respectively ten hours (mean) per week for their project. More than one-fourth (75 percentile) of the subjects contribute more than 10 hours. This is certainly an enormous effort.

The most active project member works on average 60 hours a week for the project. This subject is partly paid for his contribution. However, out of the ten most active project members in the study only three are somehow paid.

The above reported measures of central tendency do not change considerably if paid project members are excluded. The arithmetic mean of non-paid project members is nine hours. That there is almost no difference between non-paid project members and the whole sample is caused by the low number of paid project members. The median (20 hours) and the arithmetic mean (22 hours) of paid project members are considerably higher.

Hours of work per week	Frequency[a] Kuster	Cumulative Frequency[a] Kuster	Frequency FLOSS	Frequency FLOSS-US	Frequency FLOSS-JP	Frequency FLOSS-Asia	Frequency WIDI
> 2 hours	15.48%	15.48%	22.51%	45.58%	36.50%	22.60%	34.61%
2-5 hours	36.77%	52.26%	26.13%		25.20%	27.10%	
6-10 hours	21.94%	74.19%	20.89%	23.80%	15.80%	18.00%	31.61%
11-20 hours	12.90%	87.10%	14.26%	16.63%	10.40%	15.80%	18.50%
21-40 hours	10.32%	97.42%	9.14%	10.86%	6.90%	10.50%	9.96%
< 40 hours	2.58%	100.00%	7.07%	3.06%	5.20%	6.00%	5.33%
N	155	155	2461	1588	547	138	5233

[a] The number of missing values is 22.

Figure 140: Hours of work for FOSS per week[667]

A comparison between the working hours for FOSS projects in this study and the FLOSS study shows some interesting similarities and two differences.[668] The overall distribution of working hours is in both studies similar. In this study, however, fewer subjects work less than two hours. At the same time the fraction of project members devoting more than forty hours is considerably higher in the FLOSS study. This suggests that the results for subjects working between two and forty hours are very robust and may reflect the true value in the population. The results for subjects who work extremely many or very few hours for their FOSS project is probably due to a sampling effect and should be taken with precaution.

In the FLOSS-US study the subjects worked on average 11 (mean) respectively 7 (median) hours a week for their project (David, Waterman, & Arora 2003b: 36) that is somewhat more than in this study.

In the BCG study the mean is 8 (mean) respectively 3 (median) hours for all subjects (Lakhani & Wolf 2005: 10). This is somewhat lower than in this study. A major difference exists for paid FOSS project members. The average working hours of paid developers is in this study with 22 hours (mean) considerably higher than the 10 hours (mean) in the BCG study. Because the sample size of paid FOSS developers in this study is quite small these results should be noticed with a pinch of salt.

Paid project members

In Figure 141 it can be recognized that 9% of projects members are to some extent paid. Compared to other studies this is a very low fraction.[669]

---

[667] David, Waterman, and Arora (2003a), Ghosh et al. (2002c), Mitsubishi Research Institute (2004a), Mitsubishi Research Institute (2004b), and Robles et al. (2001a: 37).

[668] While this study asked about spent hours for the focal FOSS project many other studies asked about the spent hours for FOSS in general. As the BCG study showed this distinction may make a difference (Lakhani & Wolf 2005: 10). Conclusion from a corresponding comparison must be therefore taken with suspiciousness.

[669] In the BCG study 13% (Lakhani & Wolf 2005: 9), in the study of Hars and Ou (2002: 31) 16%, in the WIDI study 21% (Robles et al. 2001a: 33), in the FLOSS-JP study 27% (Mitsubishi Research Institute 2004d), in the FLOSS-US study 41% (David, Waterman, & Arora 2003a), in the FreeBSD study of Jørgensen (2000; 2001) 43%, in the study of Hertel

Payment status	Frequency	Frequency
To some extent paid	7.91%	8.70%
Not paid	83.05%	91.30%
Missing	9.04%	-
N	177	161

Figure 141: Payment status (dummy)

Salaried project members are paid between 1 and 40 hours a week (see Figure 142). One-fourth of the paid developers are paid half time or less (see Figure 143). Less than one-tenth is paid between half and three-quarters of the devoted time. Half of the paid developers are fully or almost fully paid.[670] Interestingly, one-sixth of project members are paid more than they actually work for their FOSS project. However, because the number of paid project members is very small the corresponding interpretation is not very reliable.

Payment status	Frequency	Frequency	Cumulative Frequency
0 hours	83.05%	-	-
1-4 hours	2.82%	35.71%	35.71%
5-9 hours	0.56%	7.14%	42.86%
10-29 hours	1.13%	14.29%	57.14%
30-40 hours	3.39%	42.86%	100.00%
Missing	9.04%	-	-
N	177	14	14

Figure 142: Payment status

Payed hours in % of worked hours	Frequency		Cumulative Frequency
0 % paid	76.84%	-	-
1-25 % paid	0.56%	7.69%	7.69%
26-50 % paid	1.13%	15.38%	23.08%
51-75 % paid	0.56%	7.69%	30.77%
76-100 % paid	3.95%	53.85%	84.62%
101-200 % paid	1.13%	15.38%	100.00%
Missing	15.82%	-	-
N	177	13	13

Figure 143: Paid hours in % of worked hours

The subjects who are not paid for their contribution account together for approximately 1203 hours per week (82% of total). In contrast the subjects, who are paid account together for roughly 271 hours per week (18% of total).[671] Therefore, roughly four-fifth of contributed hours are donated by non-paid developers.

et al. (2003: 1168) 43%, in the FLOSS-Asia study 45% (Mitsubishi Research Institute 2004c), and in the FLOSS study 50% of subjects are paid (Ghosh et al. 2002c). Subjectively, it is thought that such high numbers of paid developers are clearly not representative to the community. This highlights, again, the problems of proper sampling in FOSS research.

670 Most salaried contributors are paid for every hour they work for their project. Nevertheless, about one-third of paid developers are not fully paid for their contributions to FOSS. This questions the use of dummy items paid/not-paid because it is not clear whether a developer who is paid for 10% of his contributions should be considered being paid or being not paid.

671 This computation accounts for that paid developers are not always paid full-time.

Future contribution

The high number of missing values suggests that this item was difficult to answer (see chapter 11.1). From the subjects who gave a valid answer almost 50% of the subjects will neither decrease nor increase their contribution in the future (see Figure 144). Roughly 40% of the subjects will increase and about 10% of the subjects will decrease their engagement. Therefore, the engagement of the subjects as a whole will increase as it is also indicated by the item difficulty.

Contribution in the future	Frequency tables						Item analysis
	strongly decrease				strongly increase	Missing	Item difficulty[a]
	3.39%	5.65%	38.98%	24.29%	10.73%	16.95%	0.60

N=177

[a] Standardized arithmetic mean ranging from 0 (strong decrease) to 1 (strong increase)

Figure 144: Item analysis of future engagement

11.2.4 Motivation

The item difficulty for all different types of motivation is on average 0.69. The scores on motivation are therefore biased toward the right side of the scale. Interestingly, a study of Bonaccorsi and Rossi (2003: 20) with similar items observed the same effect. The maximal item difficulty is 0.91 and the minimal item difficulty is 0.50. Two items have an item difficulty above the recommended 0.8 value.

Of the 19 items on motivation 10 have an item discrimination below the recommended 0.3 value. Especially low are the items related to fun seeking, peer recognition, and pro-social motivation. Based on this analysis it can be concluded that the items related to fun seeking, peer recognition, and pro-social motivation are suspect (see also scale analysis in chapter 11.3).

Motivation		Frequency tables						Item analysis	
		strongly disagree				strongly agree	Missing	Item difficulty[a]	Item discrimination[b]
Fun seeking	Enjoy programming	1.13%	1.13%	10.73%	42.94%	40.68%	3.39%	0.81	0.18
	Code elegance	0.56%	5.08%	15.25%	35.59%	40.11%	3.39%	0.78	0.31
	Resulting code	0.56%	3.39%	15.82%	37.85%	38.42%	3.95%	0.79	0.24
Pro social	Proprietary software	15.25%	18.08%	20.34%	23.73%	18.64%	3.95%	0.53	0.27
	Ethical problem	7.91%	8.47%	15.25%	27.12%	36.72%	4.52%	0.70	0.36
	Innovation	5.08%	3.39%	11.30%	29.94%	45.20%	5.08%	0.78	0.20
	License force	0.56%	1.13%	8.47%	11.86%	72.32%	5.65%	0.91	0.23
Peer recognition	Concessions	4.52%	13.56%	37.85%	31.64%	5.08%	7.34%	0.55	0.10
	Bad Code	2.82%	11.86%	16.38%	18.64%	8.47%	41.81%	0.58	0.26
	Peer appraisal	3.39%	6.78%	25.99%	40.11%	18.64%	5.08%	0.67	0.14
Personal requirements	Fork	13.56%	20.34%	24.86%	20.34%	14.12%	6.78%	0.50	0.22
	Personal needs	3.39%	7.34%	12.43%	37.29%	33.90%	5.65%	0.74	0.40
	Fix problems	1.69%	6.21%	14.69%	41.81%	31.07%	4.52%	0.75	0.42
Reputation	Better job	14.69%	14.12%	31.64%	22.03%	12.43%	5.08%	0.51	0.51
	Talented programmers	11.30%	11.86%	27.12%	28.81%	14.12%	6.78%	0.56	0.31
	Judge skills	4.52%	18.08%	37.85%	22.60%	11.30%	5.65%	0.55	0.21
Human capital	Programming skills	0.56%	2.82%	10.17%	45.76%	36.16%	4.52%	0.80	0.36
	Feedback importance	0.00%	1.69%	18.08%	41.81%	33.33%	5.08%	0.78	0.33
	Better than course	2.26%	5.65%	16.95%	32.20%	38.42%	4.52%	0.76	0.45

N=177

[a] Standardized arithmetic mean ranging from 0 (total disagreement) to 1 (total agreement)

[b] Pearson Product Moment correlation between an item and the sum of the remaining items in a scale

Figure 145: Item analysis of motivation types

11.2.5 Business interests

For about 30% of all subjects there is a relationship between project participation and business interests (see Figure 146). From these subjects one-third (or 9% of all subjects) is paid for project participation by an employer and two-thirds (or 21% of all subjects) have an own business related to the project. Therefore, the quantity of self-employed contributors with business interests outnumbers the quantity of employed contributors twice. However, employed developers contribute on average 24 hours a week (mean) while self-employed developers contribute “only” 10 hours on average. Taking this into account the magnitude of contributed hours per week of employed and self-employed developers is almost similar.[672]

Roughly 9% of all subjects state in item 45 that they are paid (see above). Almost exactly the same fraction of subjects stated in item 24 that they are paid or partly paid for project participation (see 11.2.3). This indicates that the items related to the status of being paid are highly reliable.

Business interest					
Business interest: total	Buisness interest: paid[a]	Buisness interest: own business[a]	No relatinship	Missing	N
27.70%	8.50%	19.20%	66.10%	6.20%	177
29.50%	9.00%	20.50%	70.50%	-	166

[a] Included in total

Figure 146: Business interests

Figure 147 informs why subjects with a business interest contribute to FOSS projects. To ensure that the application provides demanded functionalities and that bugs get fixed are the main reasons why subjects respectively their employees invest scare resources for FOSS participation. The ability to fork the project and the reputation in the FOSS community are significantly less important (see confidence intervals in Figure 148).

Business interest			
Mean	Business interest: total	Buisness interest: paid[a]	Buisness interest: own business[a]
Functionalities	4.06	4.60	3.82
Bug fixing	4.16	4.67	3.94
Fork	3.48	3.47	3.48
Reputation	3.53	3.53	3.53

[a] Included in total

Figure 147: Reason for business interests

[672] All self-employed developers of the survey contribute together 306 hours a week to FOSS projects while employed developers contribute together 333 hours.

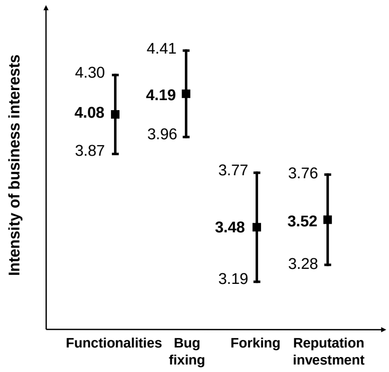


Figure 148: Reasons for business interests: 90% confidence interval

11.2.6 Basic need satisfaction

The item difficulty for all basic needs is on average 0.68. The items are therefore biased toward the right side of the scale. The maximal item difficulty is 0.85 and the minimal item difficulty is 0.54. One item has an item difficulty above the recommended 0.8 value. Of the 9 items 8 items have an item discrimination below the recommended 0.3 value. Based on this item analysis it can be concluded that the items related to basic need satisfaction are poor (see also scale analysis in chapter 11.3). The items related to relatedness are almost acceptable.

	Basic Need Satisfaction	Frequency table						Item analysis	
		strongly disagree				strongly agree	Missing	Item difficulty[a]	Item discrimination[b]
Autonomy	Influence patch merging	2.82%	10.73%	25.99%	32.20%	18.64%	9.60%	0.64	-0.08
	Pressure use tools	7.34%	18.08%	19.77%	24.86%	20.34%	9.60%	0.59	-0.10
	Express ideas	0.56%	0.00%	7.34%	37.85%	47.46%	6.78%	0.85	0.27
Competence	Programmers praise me	1.13%	7.34%	31.07%	41.24%	10.17%	9.04%	0.64	0.17
	Sense of accomplishment	1.69%	2.82%	37.85%	35.03%	14.12%	8.47%	0.65	0.19
	Show capabilities	2.82%	10.17%	27.12%	27.68%	24.86%	7.34%	0.66	0.03
Relatedness	Communication	1.69%	5.65%	22.60%	32.77%	31.07%	6.21%	0.72	0.17
	Programmers would miss me	6.78%	18.64%	30.51%	24.86%	11.30%	7.91%	0.54	0.26
	Sympathy for programmers	0.56%	1.13%	17.51%	39.55%	33.90%	7.34%	0.78	0.38

N=177

[a] Standardized arithmetic mean ranging from 0 (total disagreement) to 1 (total agreement)

[b] Pearson Product Moment correlation between an item and the sum of the remaining items in a scale

Figure 149: Basic need satisfaction

11.2.7 Demographics

Age

The majority of subjects is between 20 and 29 years old (see Figure 150). The second largest group is between 40 and 49 years old. There are not many subjects younger than 20 years or older than 40 years. The average age of subjects is 28 years (arithmetic mean) respectively 27 years (median). The oldest subject is 55 while the youngest subject is 16 years old.

Age	Frequency	Cumulative Frequency
> 20	6.78%	7.19%
20-29	54.24%	64.67%
30-39	24.86%	91.02%
40-49	7.34%	98.80%
< 50	1.13%	100.00%
Missing	5.65%	-
N	177	167

Figure 150: Age

Age	Frequency	Cumulative Frequency
> 20	9.29%	9.55%
20-29	60.78%	72.03%
30-39	22.07%	94.72%
40-49	4.01%	98.84%
< 50	1.13%	100.00%
Missing	2.72%	
N	5487	5338

Figure 151: Age in the WIDI study (own computation)

Compared to the WIDI study in which the subjects were 27 years (arithmetic mean) respectively 25 years old (median) the subjects of this study are slightly older (Robles et al. 2001a: 24-25 and own computation). Otherwise, the distribution pattern of this study is very similar to the one of the WIDI study (see Figure 150 and Figure 151).

The subjects of the FLOSS study, the FLOSS-US study, the FLOSS-Asia study, and the study conducted by Luthiger had about the same average age (see Figure 152). Subjects of the BCG study and the FLOSS-JP study were on the average somewhat older. It can be summarized that the average age as well as the distribution of ages in this study is similar to other studies.

	Average age								
	Kuster	FLOSS	FLOSS-US	FLOSS-JP	FLOSS-Asia	WIDI	BCG	Hertel et al.	Luthiger
Median	27	26	27	31	27	25	[a]	[a]	27
Mean	28.4	27.1	[a]	31.2	27.9	26.8	30	30	28.7
N	167	2774	1588	547	138	5487	684	141	1330

[a] not reported

Figure 152: Comparison of average age[673]

Programming experience

The surveyed subjects have a high programming experience (see Figure 153). On average the subjects have an experience of 15 (mean) respectively 14 (median) years. The subject with the lowest experience codes for 5 years while the subject with the highest experience codes for 44 years. On average the subjects in the BCG study had with a mean of 12 years less programming experience than subjects in this study (Lakhani & Wolf 2005: 9).

Years of programming experience	Frequency	Cumulative Frequency
5-9	9.04%	9.82%
10-14	38.42%	51.53%
15-19	33.90%	88.34%
20-24	7.34%	96.32%
26-44	3.39%	100.00%
Missing	7.91%	-
N	177	163

Figure 153: Programming experience

If the programming experience is put in relation to project experience one finds surprisingly that subjects have on average 12 years of programming experience (mode, median, and mean) *before* they have started to contribute to the FOSS project in question. There are only four subjects who started to contribute to a FOSS project without any prior programming experience. One-fourth of the subjects have a programming experience between zero and 8 years before they contribute to a FOSS project, one-fourth an experience between 8 and 12 years, one-fourth an experience between 12 and 14 years, and one-fourth of the subjects has a prior

[673] David et al. (2003b: 16), Ghosh et al. (2002a: 8), Hertel et al. (2003: 1167), Lakhani and Wolf (2005: 8-9), Luthiger (2006: 183-184), Mitsubishi Research Institute (2004a; 2004b), and Robles et al. (2001a: 24-25).

programming experience between 14 and 43 years. It can be concluded that contributors have a high prior programming experience before they contribute to a FOSS project.

Working hours

On average developers work 40 hours a week in a paid job. Roughly 29% of the subjects work 41 or more hours per week (see Figure 154). Almost one-fourth of the subjects work exactly 40 hours. The arithmetic mean is 31 hours that is heavily influenced by one-fifth of the subjects who do not work at all in a paid job. If students are excluded from the analysis the arithmetic mean is 40 hours. The subject with the highest workload works 110 hours a week.

Hours of work per week	Frequency	Cumulative Frequency
0 hours	18.08%	20.38%
1-10 hours	5.65%	26.75%
11-20 hours	6.78%	34.39%
21-30 hours	5.08%	40.13%
31-40 hours	27.68%	71.34%
41-50 hours	17.51%	91.08%
< 50 hours	7.91%	100.00%
Missing	11.30%	-
N	177	157

Figure 154: Hours of work per week

Education

In the study 90% of the subjects have an education above compulsory education (see Figure 155). More than four-fifth of the subjects has completed the entry requirement for university and almost 60% have some sort of university degree. The educational level of the subjects in this study is therefore very high. In 2005 the average upper secondary graduation rate (ISCED 3A: programs designed to prepare for direct entry to tertiary-type A education) of all OECD countries is 59% (OECD 2007).[674]

[674] However, see the education level of computer programmers in the US below.

Highest completed education	Frequency	Cumulative Frequency
Compulsory education	9.04%	100.00%
Above compulsory education	5.65%	90.18%
University: entry requirement	23.16%	84.05%
University: Bachelor	21.47%	58.90%
University: Master	29.94%	35.58%
University: PhD	2.82%	3.07%
Missing	7.91%	-
N	177	163

Figure 155: Highest completed education

In the FLOSS study roughly 70% of the subjects have received a university degree (Ghosh et al. 2002c) that is somewhat more than in this study (see Figure 156). While the rate of master degrees is similar the rates of Ph.D. and Bachelor degrees are higher in the FLOSS study. In the FLOSS-US study 73% (David, Waterman, & Arora 2003b: 9), in the FLOSS-JP study 85% (Mitsubishi Research Institute 2004b), and in the FLOSS-Asia study 69% (Mitsubishi Research Institute 2004a) have some sort of university degree. It can be concluded that the education level of the subjects in this study is below comparable studies.

Highest completed education						
	Kuster	FLOSS	FLOSS-US	FLOSS-JP	FLOSS-Asia	WIDI
Below Bachelor / College	41.1%	29.6%	19.4%	31.2%	15.0%	37.2%
Bachelor / College	23.3%	32.6%	38.0%	35.2%	63.9%	17.4%
Master	32.5%	28.1%	36.7%	27.1%	15.8%	40.6%
PhD	3.1%	9.8%	5.9%	6.5%	5.3%	4.8%
Some kind of university degree	58.9%	70.4%	80.6%	68.8%	85.0%	62.8%
N	163	2143	1498	542	133	4872

Figure 156: Education in different FOSS studies

Two-thirds of subjects in this study who do not study and most probably have completed[675] their formal education have some sort of university degree (see Figure 157). This is comparable to the fraction of computer programmers in the US in 2004 who have at least a bachelor degree (67.4%, BLS 2006a). Interestingly, while

[675] It may be that there are subjects who have interrupted their studies. However, such cases are most probably not very frequent.

the fraction of FOSS developers holding a bachelor degree is similar to all computer programmers in the US the fraction of FOSS developers holding a master degree is by far higher.[676]

Highest completed education	Frequency	Cumulative Frequency
Compulsory education	7.07%	100.00%
Above compulsory education	7.07%	92.86%
University: entry requirement	19.19%	85.71%
University: Bachelor	21.21%	66.33%
University: Master	40.40%	44.90%
University: PhD	4.04%	4.08%
Missing	1.01%	-
N	99	98

Figure 157: Highest completed education of subjects who do not study

Students

More than one-third of the subjects are students (Figure 158). Approximately two-thirds of these students study something related to IT.

State of current education		Frequency		Frequency (without missing)	
Student	IT related	34.46%	23.73%	38.13%	26.25%
	Non-IT related		10.73%		11.88%
No student		55.93%	55.93%	61.88%	61.88%
Missing		9.60%	9.60%	-	-
N		177		160	

Figure 158: Current education

In this study the fraction of students is higher than in the WIDI study (30%, own computation), the FLOSS-US study (29%, David, Waterman, & Arora 2003b: 10), the FLOSS study (21%, Ghosh et al. 2002c)[677], the BCG study (20%, Lakhani & Wolf 2005: 9), the FLOSS-Asia study (15.8%, Mitsubishi Research Institute 2004c), and

[676] A reason for this may be that in Europe subjects going to university strive until recently almost always for a master but not for a bachelor degree.

[677] There is another item in which subjects could indicate that they are a student (see Ghosh et al. 2002c). However, the raw data for this item seems to be incorrect because the data does neither add up to 100% nor does it completely correspond to the figures in the final report (see Ghosh et al. 2002a).

the FLOSS-JP study (14.4%, Mitsubishi Research Institute 2004d). Therefore, it is obvious that more students participated in this study than in other studies. This may be explained by the fact that the two conferences at which the subjects were surveyed took place within university buildings. In university buildings there are of necessity more students than in online communities that can be accessed with less effort.

Occupation

The professional life or the studies of almost one-fifth of the subjects have no IT relationship. This is similar to other studies in which the fraction of subjects with a professional relationship to IT ranges between 15% and 28% (Ghosh et al. 2002c; Mitsubishi Research Institute 2004c, 2004d; Robles et al. 2001a: 31). The occupation or studies of another fifth of the subjects are related to software *and* hardware. Only a tiny fraction has only a relationship to hardware. The occupations or studies of more than half of the subjects are related only to software.

Most subjects with a relationship to IT (software or/and hardware) are concerned with development (62%). Subjects with a relationship to IT are also often concerned with administration (39%), support (23%), or they are students (29%). Less frequently subjects have a relationship to sales (12%) or they are employed as a professor, lecturer, assistant, or teacher (9%). Subjects whose occupation is only related to software are more frequently connected to development compared to subjects whose occupation is only related to hardware. Subjects whose occupation is only related to hardware are more frequently connected to sales. However, the number of subjects only related to hardware is too small to draw any sound conclusion.

English proficiency

The item difficulty (0.75) suggests that the subjects consider their English proficiency as good (see Figure 159). A vast majority of 77% who answered the question consider their English as good or very good.

English proficiency	**Frequency tables**						**Item analysis**
	Not very good				**Very good**	**Missing**	**Item difficulty[a]**
English proficiency	1.13%	4.52%	14.12%	38.42%	27.68%	14.12%	0.75

N=177

[a] Standardized arithmetic mean ranging from 0 (not very good) to 1 (very good)

Figure 159: English

Subjects of the Chemnitz sample (item difficulty 0.71) have a lower English proficiency than subjects of the Brussels sample (item difficulty 0.80). Observation of subjects in Brussels suggests that their statements concerning English proficiency are correct. The conference language was English and participants showed that they are familiar with it. Anecdotal evidence from Chemnitz and observation of subjects filling out surveys point to the presumption that the corresponding subjects overrated their English proficiency on the average.

In the FLOSS-JP study only 6% of developers declare that they can speak fluently English and write without a dictionary (Mitsubishi Research Institute 2004b).[678] In the FLOSS-Asia study the fraction of developers with the same level of English proficiency is 26% (Mitsubishi Research Institute 2004a). It can be concluded that the English proficiency of the subjects in this study is higher than the one in the FLOSS-Asia or the FLOSS-JP study.

Gross income

Approximately half of the subjects in the sample do have a monthly gross income above and below 2000 US$ (see Figure 160). A tiny fraction earns more than 7000 US$ while one-fourth earns less than 500$.

Monthly gross income	**Frequency**	**Cumulative Frequency**
< 500 US$	16.95%	24.39%
501-2000 US$	18.64%	51.22%
2001-4000 US$	23.16%	84.55%
4001-7000 US$	7.91%	95.93%
> 7000 US$	2.82%	100.00%
I prefer not answering the question	19.77%	-
	10.73%	-
N	177	123

Figure 160: Monthly gross income

The median gross income of the subjects in this study and of the subjects in the FLOSS study is similar (Ghosh et al. 2002a: 15, see also chapter 2.4). In both studies approximately one-half of the subjects earns more than 2000 US$ while the other half earns less. In the FLOSS-JP study and in the WIDI study the median gross income is considerably higher (Mitsubishi Research Institute 2004d; Robles et al. 2001a: 37).

Email

Roughly 40% of subjects asked to get an email with the results (see Figure 161).

[678] There were five different categories and this category indicated the highest English proficiency.

Email provided	Frequency	Frequency (without missing)
Yes	37.85%	40.36%
No	55.93%	59.64%
Missing	6.21%	-
N	177	166

Figure 161: Email provided

Comments

A minority of 14% provided an individual comment about their motivation to participate in FOSS projects (see Figure 162).

Comments provided	Frequency	Frequency (without missing)
Yes	12.99%	13.86%
No	80.79%	86.14%
Missing	6.21%	-
N	177	166

Figure 162: Comments

In the following, some of the comments are presented:

- “The software I am developing was completely missing on Linux before. That's why 5 decided to develop free Linux solutions … . FOSS is needed for a fair and justice world!”
- “Share your work is the best way to work! (I don't talk about earning money)”
- “Main motivation: Own need for features / bug fixes. See users happy that their needs are also satisfied.”
- “It is a lousy job, but someone has to do it.”
- „It's fun :)“
- “I have 4 words to say: I love this community. (Google for Steve Ballmer)” (see Figure 163). This is an allusion of a famous video of Microsoft’s CEO Steve Ballmer at an employee convention. The video features him dancing and screaming what gave the performance the name “Dance Monkey”. The phrase “I love this community” is a modification of Ballmer’s statement “I love this company”.

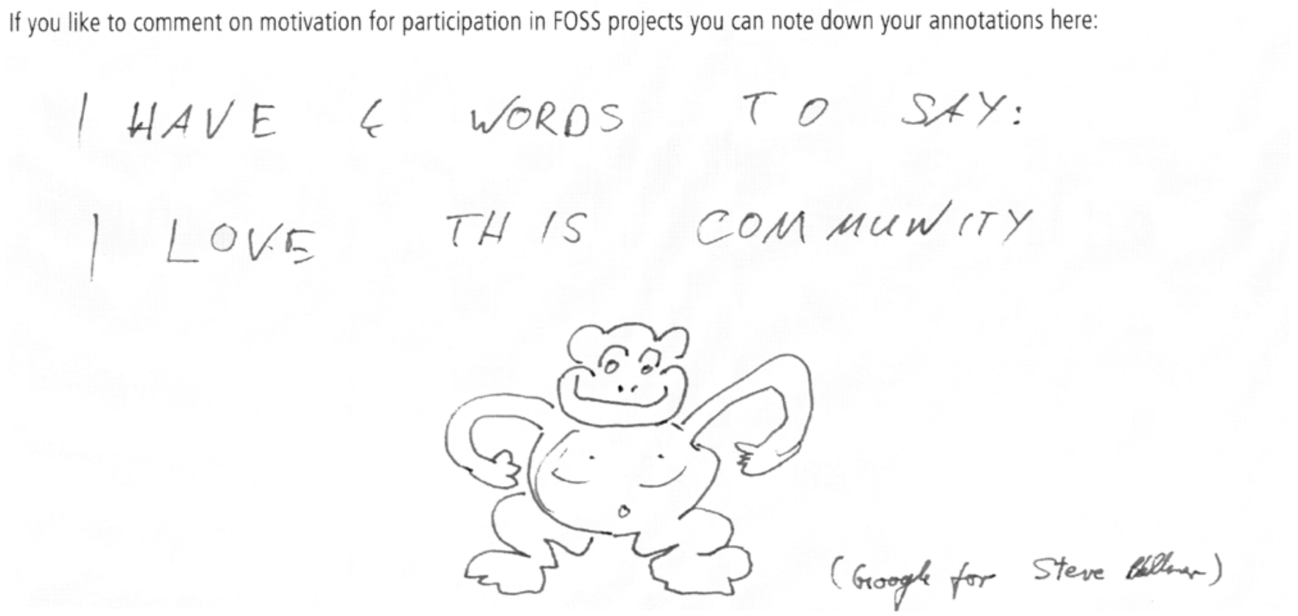


Figure 163: Example of an individual comment

- “I feel it's important that I have the choice as to what I can do on my computer. Being limited by companies with proprietary OS's + products is not acceptable to me. It's my m/c, I can code what I want.”
- “Give back because I received a lot”
- “Free software, Free society (as I just saw on a T-shirt)”
- “An itch that needed to be scratched”

11.3 Scale analysis

There are several different approaches to deal with latent variables (Aaker & Bagozzi 1979: 157). One approach is to build a scale based on relevant items. The scale can be a simple sum or involving some weighting. In the following, scales are built simply by adding the values of the different items.

Items should only be included in a scale if they measure the construct appropriately (Churchill 1979: 68). Below it is discussed which of the items are appropriate and therefore included in the scale.

11.3.1 Dimensionality and internal consistency

In classical test theory it is assumed that reflexive constructs are unidimensional respectively that they have high internal consistency (see Anderson & Gerbing 1982: 454-455; Diamantopoulos & Siguaw 2006: 265). Basically, unidimensionality and internal consistency are logically similar but they are measured differently. Dimensionality can be investigated by performing a factor analysis (Churchill 1979: 69; Comrey 1988; Diamantopoulos & Siguaw 2006: 269). The same items of a construct should have substantial loadings on the same factor (Comrey 1988: 755). The number of dimensions underlying the construct should agree with those conceptualized. (Churchill 1979: 69). Internal consistency can be measured by Cronbach alpha (Aaker & Bagozzi 1979: 156) that has already been discussed in chapter 9.7.2.

For formative constructs it is not appropriate to test for dimensionality and internally consistency. Only reflective but not formative models assume unidimensionality (Diamantopoulos & Siguaw 2006: 265) and internal consistency. This obsoletes corresponding tests for formative models. Tests on internal consistency and dimensionality can neither theoretically be applied to formative models nor interpreted meaningfully. However, for convenience the corresponding results are also reported for formative models. The related results are though not interpreted.

11.3.1.1 Motivation scales

Fun seeking

The original fun seeking scale includes the three items “enjoy programming”, “code elegance”, and “resulting code”. As can be seen in Figure 164 a factor analysis produces only one factor with an eigenvalue above one. The fun seeking scale is therefore unidimensional and there is no reason to exclude an item. However, the factor loading of the item “enjoy programming” is lower than the one of the other two items indicating that the relationship between this item and the fun seeking scale is less tight.

Cronbach alpha of the fun seeking scale improves slightly if “enjoy programming” is excluded which suggests to delete this item. Consequently, the item “enjoy programming” is excluded for the final fun seeking scale.

Motivation		Scale anlysis								
		Dimensio-nality (factor loading)[a/b]	Cronbach alpha[c]	Cronbach alpha if item deleted[c]	Correlation[d/e]		A	B	C	D
Fun seeking	Enjoy programming[g]	0.57	0.406	**0.414**	A	Correlation	/	0.19	0.09	
						Significance		***0.01***	*0.11*	
	Code elegance	0.77		0.17	B	Correlation	0.19	/	0.26	
						Significance	***0.01***		***0.00***	
	Resulting code	0.68		0.32	C	Correlation	0.09	0.26	/	
						Significance	*0.11*	***0.00***		
Pro social	Proprietary software[f]	0.65	**0.45**	0.38	A	Correlation	/	0.28	0.06	0.21
						Significance		***0.00***	*0.21*	***0.00***
	Ethical problem[f]	0.74		0.26	B	Correlation	0.28	/	0.24	0.17
						Significance	***0.00***		***0.00***	***0.01***
	Innovation[f]	0.51		0.44	C	Correlation	0.06	0.24	/	0.09
						Significance	*0.21*	***0.00***		*0.13*
	License force[f]	0.56		0.42	D	Correlation	0.21	0.17	0.09	/
						Significance	***0.00***	***0.01***	*0.13*	
Peer recognition	Conscessions[g]	-0.02 0.93	0.29	**0.34**	A	Correlation	/	0.16	-0.03	
						Significance		***0.06***	*0.36*	
	Bad Code	0.67 0.43		-0.02	B	Correlation	0.16	/	0.21	
						Significance	***0.06***		***0.02***	
	Peer appraisal	0.85 -0.19		0.26	C	Correlation	-0.03	0.21	/	
						Significance	*0.36*	***0.02***		
Personal requirements	Fork[f]	0.52	**0.52**	0.64	A	Correlation	/	0.18	0.20	
						Significance		***0.01***	***0.01***	
	Personal needs[f]	0.81		0.32	B	Correlation	0.18	/	0.45	
						Significance	***0.01***		***0.00***	
	Fix problems[f]	0.82		0.31	C	Correlation	0.20	0.45	/	
						Significance	***0.01***	***0.00***		
Reputation	Better job	0.86	0.52	0.09	A	Correlation	/	0.39	0.29	
						Significance		***0.00***	***0.00***	
	Talented programmers	0.71		0.47	B	Correlation	0.39	/	0.04	
						Significance	***0.00***		*0.31*	
	Judge skills[g]	0.55		**0.59**	C	Correlation	0.29	0.04	/	
						Significance	***0.00***	*0.31*		
Human capital	Programming skills	0.71	**0.56**	0.49	A	Correlation	/	0.21	0.38	
						Significance		***0.00***	***0.00***	
	Feedback importance	0.68		0.52	B	Correlation	0.21	/	0.33	
						Significance	***0.00***		***0.00***	
	Better than course	0.80		0.34	C	Correlation	0.38	0.33	/	
						Significance	***0.00***	***0.00***		

[a] Factor loadings extracted with principal component analysis / only factors with an eigenvalue above one
[b] Factors of multidimensional scales are Varimax rotated
[c] Cronbach alpha of the final scale (after item deletion) are bold
[d] Pearson product-moment correlation
[e] Significant correlations (one-sided, P =< 0.1) are bold
[f] Part of a formative model
[g] Item excluded in the scale to increase Cronbach alpha

Figure 164: Scale analysis of motivation

Pro-social motivation

The pro-social behavior scale consists of the items “proprietary software”, “ethical problem”, “innovation”, and “license force”. A discussion about excluding items is obsolete because the scale is based on a formative model. In the final pro-social motivation scale all items are included.

Peer recognition

The original peer recognition scale consists of the three items “concessions”, “bad code”, and “peer appraisal”. Scale analysis shows that the original scale has two dimensions (see Figure 164). The items “bad code” and “peer appraisal” are related to one factor and the item “concessions” to another factor. Making concessions about source code to please other developers seems therefore not to be related to peer recognition. This suggests excluding the item “concessions”.

Excluding the item “concessions” is also suggested by the Cronbach alpha analysis. Internal consistency increases if this item is deleted. The final peer recognition scale consists therefore of “bad code” and “peer appraisal”.

Even after the deletion of the item “concessions” the scale remains very weak. Next to the problems related to the wording of the item “concessions” there seems to be another reason for the bad performance of the scale. The high number of missing values for the item “bad code” (see chapter 11.1) contributes most probably also to the poor performance of the scale.

Personal requirement

The personal requirement scale consists of the items “fork”, “personal needs”, and “fix problems”. A discussion about excluding items is obsolete because the scale is based on a formative model. In the final personal requirement scale all items are included. However, it is interesting to note that forking has a lower factor loading. This indicates that the ability to “fork” is different from “personal needs” and “fix problems”.

Reputation

The original reputation scale with the three items “better job”, “talented programmers”, and “judge skills” is unidimensional (see Figure 164). The factor loading for the item “judge skill” is the lowest indicating that it has the weakest relationship to the reputation scale.

If the item “judge skills” is excluded the Cronbach alpha increases. To increase internal consistency this item is consequently excluded from the reputation scale.

Human capital

The items “programming skills”, “feedback importance”, and “better than course” are part of the human capital scale. The original scale with all three items is unidimensional and excluding an item does not increase Cronbach’s alpha (see Figure 164). Therefore, the final human capital scale is the same as the original one.

All motivations belonging to a reflexive construct

A further test of the four reflexive motivation scales is a factor analysis that includes all corresponding items included in the scales. Each of the extracted components should clearly be associated with one of the four motivation scales. The factor loadings of each item should be positively related to the factor loadings of the others items of the same motivation scale. As can be seen in Figure 165 the factor loadings of the items of the same motivation scale are, overall, heavily related to each other. This suggests that the motivation scales are usable for further statistical analysis. This is backed up by the observation that the items included in the reflexive motivation scales correlate significantly with the other items of the corresponding scale (see Appendix II).[679] However, the fun seeking scales performs not satisfactory.

Rotated Component Matrix[ab]

Item	Fun seeking	Peer recognition	Reputation	Human capital
Code elegance	**0.87**	0.11	-0.01	0.15
Resulting code	**0.22**	0.57	0.05	0.28
Bad Code	0.51	**0.60**	-0.12	-0.23
Peer appraisal	-0.02	**0.70**	0.37	0.03
Better job	-0.10	0.12	**0.79**	0.22
Talented programmers	0.07	0.04	**0.85**	0.02
Programming skills	0.11	0.19	0.17	**0.76**
Feedback importance	-0.38	0.54	-0.26	**0.52**
Better than course	0.03	-0.04	0.11	**0.82**

N=177
[a] Components extracted with principal component analysis
[b] Varimax rotated

Figure 165: Factor analysis of reflexive motivation scales

11.3.1.2 Basic need scales

Autonomy

The original autonomy scale consists of the three items “influence patch merging”, “pressure use tools”, and “express ideas”. Scale analysis shows that the original scale has two dimensions (see Figure 166). The items “influence patch merging” and “express ideas” are related to one factor and the item “pressure use tools” to another factor. This suggests excluding the item “pressure use tools”. To do so is also suggested by the Cronbach alpha analysis. It increases if the item is deleted.

[679] See also Cronbach alpha that basically depends on correlations between the items of a scale.

The final autonomy scale consists of the items “influence patch merging” and “express ideas”. Even after the deletion of the item “pressure use tools” the scale remains very weak.

Basic needs		Scale anlysis								
		Dimensionality (factor loading)[a/b]		Cronbach alpha[c]	Cronbach alpha if item deleted[c]	Correlation[d/e]				
								A	B	C
Autonomy	Influence patch merging	-0.62	0.61	-0.02	0.19	**A**	Correlation	/	-0.22	0.23
							Significance		***0.00***	***0.00***
	Pressure use tools[f/g]	0.88	0.18		**0.34**	**B**	Correlation	-0.22	/	0.13
							Significance	***0.00***		***0.06***
	Express ideas	0.17	0.89		-0.55	**C**	Correlation	0.23	0.13	/
							Significance	***0.00***	***0.06***	
Competence	Programmers praise me	0.79		0.23	0.06	**A**	Correlation	/	0.27	0.03
							Significance		***0.00***	*0.35*
	Sense of accomplishment	0.79			0.04	**B**	Correlation	0.27	/	0.04
							Significance	***0.00***		*0.31*
	Show capabilities[f/g]	0.16			**0.41**	**C**	Correlation	0.03	0.04	/
							Significance	*0.35*	*0.31*	
Relatedness	Communication[f/g]	0.53		0.43	**0.49**	**A**	Correlation	/	0.09	0.22
							Significance		*0.13*	***0.00***
	Programmers would miss me	0.72			0.34	**B**	Correlation	0.09	/	0.34
							Significance	*0.13*		***0.00***
	Sympathy for programmers	0.80			0.16	**C**	Correlation	0.22	0.34	/
							Significance	***0.00***	***0.00***	

[a] Factor loadings extracted with principal component analysis / only factors with an eigenvalue above one
[b] Factors for multidimensional scales are Varimax rotated
[c] Cronbach alpha of the final scale (after item deletion) are bold
[d] Pearson product-moment correlation
[e] Significant correlations (one-sided, P =< 0.1) are bold
[g] Item [f] Question excluded in the scale to increase Cronbach alpha
[g] Negatively worded questions

Figure 166: Scale analysis of basic need satisfaction

Competence

The original competence scale consists of the three items “programmers praise me”, “sense of accomplishment”, and “show capabilities”. The competence scale is unidimensional (see Figure 166). The scale analysis shows that Cronbach alpha increases considerably if the item “show capabilities” is deleted. Consequently, this item is deleted.

The final competence scale consists of the items “programmers praise me” and “sense of accomplishment”.

Relatedness

The original relatedness scale consists of the three items “communication”, “programmers would miss me”, and “sympathy for programmers”. The relatedness scale is unidimensional (see Figure 166). Although Cronbach alpha increases only slightly the item “communication” is deleted.

The final relatedness scale consists of the items “programmers would miss me” and “sympathy for programmers”.

Negatively worded items

As one can see in Figure 166 all three negatively worded items “pressure use tools”, “show capabilities”, and “communication” correlate very badly to the corresponding items of the same scale. If these three items are excluded Cronbach alpha increases considerably. It may be assumed that the negative wording confused subjects. Dudycha and Carpenter (1973: 120) concluded that negatively worded items are more difficult to answer than positive ones. The related problems are probably multiplied by the fact that there were only few negatively worded items in this survey and that they appeared not until the middle of the survey. Because of their experience with the first items subjects may have thought that there are only positively worded items that may have caused something similar to a sequence effect (Huck & Bowers 1972). This may have directed subjects to not reading items thoroughly anymore.

11.3.2 Axioms of classical test theory

In the following, the three axioms of classical test theory (Novick 1966) objectivity, reliability, and validity (see chapter 9.7) are discussed in relation to the motivation and the basic need scales.

11.3.2.1 Objectivity

The conducted survey is self-administered. The subjects choose the appropriate answer themselves. The researcher has therefore no room for interpretation. The only exception is the very last item that asks to write down annotations on the motivation to join a FOSS project. However, this item is not analyzed systematically. It is mainly used for finding anecdotal evidence and to control whether major motivation types are covered by the survey. Therefore, it can be assumed that the data in this study is completely objective (see chapter 9.7.1).

11.3.2.2 Reliability

The reliability (see chapter 9.7.2) of the motivation scales is between 0.34 and 0.59 (see chapter 11.3.1.1). The median reliability is 0.49. The reliability of the basic need scales is between 0.34 and 0.49 and the median reliability is 0.41 (see chapter 11.3.1.2).

It was stated in chapter 9.7.2 that a reliability of 0.7 is in most cases sufficient. A reliability below 0.7 was considered acceptable for special research purposes. Unfortunately, overall reliability is far below what seems to be sufficient. Therefore, the results found in this study must be handled carefully. Especially error-prone are the peer recognition scale and the autonomy scale. The corresponding reliability is poor.

The reasons for the low reliability may be manifold. The low number of items is certainly an important reason. The mother tongue of most subjects was not English that may also have compromised internal consistency. This reasoning is supported by the fact that reliability increases to 0.74 for reputation, to 0.76 for human capital, and to 0.67 for peer recognition if it is only computed for subjects who stated that their English is very good.[680] For the same subpopulation reliability decreases for the fun seeking scale. Nevertheless, linguistic defects seem to have been a major problem for reliability. Clearly, the most obvious explanations that the i) items are inaccurate, ii) that the items do not capture the construct properly, or iii) that the construct is not completely unidimensional cannot be excluded.

11.3.2.3 Validity

There are no other tests in FOSS research to corroborate the scales of this survey. This excludes the possibility to analyze criterion validity (see chapter 9.7.3). Construct validity were possible to compute but would be too complicated and laborious. Content validity is therefore the only passable possibility for validation. This was done by developing the items with a member of the FOSS community. Additionally, a first version of the survey was discussed with a FOSS group at the Swiss Federal Institute of Technology in Zurich.

11.4 Results: Inferential statistics

In the following, the hypotheses specified in chapter 8.2 are tested. First of all, hypotheses related to motivation are analyzed. Afterwards, hypotheses connected to motivation and governance structures respectively mechanisms are discussed. The interactions of motivation and governance structures respectively mechanisms are then researched in order to explain satisfaction with participation in FOSS development. Hypotheses related to motivation on the one hand and the type of engagement, community affiliation, and development phases on the other hand are tested afterwards. Finally, some hypotheses related to governance structures respectively mechanisms and development phases are researched.

11.4.1 Motivation

Hypothesis 1 and 2

It was hypnotized that fun seeking and investment in human capital are the two most important types of motivation (hypothesis 1) and that investment in reputation is the least important motivation (hypothesis 2). The analysis of these two hypotheses is not very reliable. This study is based on a non-probability sample. Inferences about population characteristics such as motivation are therefore error-prone (see chapter 9.3). In consequence, the following presentation of the results must be interpreted with caution.

As can be seen in Figure 167 fun seeking and investment in human capital are the two most important types of motivation. The corresponding item difficulties for these two types of motivation are higher than the ones for the other motivation types. As can be seen by the 90% confidence intervals for fun seeking and investment in human capital this result is significant.

Investment in reputation is clearly the least important type of motivation (see Figure 167). This result is significant on the 0.1 level (see 90% confidence interval of item difficulty). In consequence, hypotheses 1 and 2 are supported.

---

[680] Reliability increases as well if only subjects from Brussels are analyzed. For the motivation scale the reliability is then between 0.54 and 0.78 while the median is 0.62.

Motivation	Motivation importance		90% confidence interval of item difficulty		BCG survey[bc]
	Ranking	Item difficulty[a]	Lower	Upper	Ranking
Fun seeking	1	0.79	0.76	0.81	1
Human capital	2	0.78	0.76	0.80	2
Pro-social	3	0.73	0.70	0.75	4
Personal requirement	4	0.67	0.64	0.69	3
Peer recognition	5	0.61	0.58	0.64	6
Reputation	6	0.53	0.50	0.57	5

[a] The item difficulty relates to the final scale

[b] To make a comparision possible, the category work related contribution is omitted.

[c] It should be noted that the methodologies of the BCG and this survey were quite different. Comparisions should be therefore interpreted carefully.

Figure 167: Motivation

Interestingly, the ranking of motivation in this study and the one in the BCG study is similar (see Figure 167).

Hypothesis 3

It was hypnotized that personal requirement is positively related to fun seeking and to investment in human capital. Indeed, personal requirement is significantly correlated to fun seeking and to investment in human capital (see Figure 168).[681] In consequence, hypotheses 3ab are supported.

Correlations between motivation scales

	Fun seeking	Pro social	Peer recognition	Personal requirement	Reputation	Human capital
Fun seeking	-	0.11 *	0.33 ****	0.30 ****	0.08	0.23 ****
Pro social	0.11 *	-	0.12	0.07	-0.01	0.12 *
Peer recognition	0.33 ****	0.12	-	0.21 ***	0.09	0.14 *
Personal requirement	0.30 ****	0.07	0.21 ***	-	0.08	0.14 **
Reputation	0.08	-0.01	0.09	0.08	-	0.10 *
Human capital	0.23 ****	0.12 *	0.14 *	0.14 **	0.10 *	-

Two-sided, Kendall-Tau-b

* Significant at the 0.1 level / ** Significant at the 0.05 level / *** Significant at the 0.01 level / **** Significant at the 0.001 level

Figure 168: Correlation table of motivation scales

[681] All significance levels reported in Figure 168 are two-sided. Therefore, the significance levels related to hypothesis 3ab are lower (0.000001 for the correlation between personal requirement and fun seeking and 0.013 for the correlation between personal requirement and human capital).

As can be seen in Figure 168 there are several motivation types significantly related to each other. At the 0.1 level the following relationships are significant:

- Fun seeking is related to all other motivation types except reputation investment.
- Pro-social behavior is, at a low level, related to fun seeking and human capital.
- Peer recognition is related to fun seeking, human capital, and personal requirement.
- Personal requirement is related to fun seeking, human capital, and peer recognition.
- Reputation investment is related to no other motivation type except human capital (but only at a very low level).
- Human capital is related to all other types of motivation.

Two of these significant correlations are recognizable: *Firstly*, investment in human capital is significantly correlated to all other motivation types. It can be presumed that investment in human capital is always an important type of motivation. *Secondly*, investment in reputation is practically not related to other types of motivation. It can be argued that FOSS hackers motivated by an investment in reputation are, at least as far as motivation is concerned, different from other hackers.

Hypothesis 4

Hypothesis 4 suggested that fun seeking, pro-social motivation, and peer recognition group while personal requirement, personal requirement, investment in reputation, and investment in human capital group as well.

A factor analysis of the six motivation types extracts two factors with an eigenvalue above 1. These two factors explain 53% of the variance. As can be seen in Figure 169 the factor loadings for the first component are high for fun seeking, peer recognition, and pro-social motivation but low for human capital, reputation, and personal requirement. In consequence, this component was interpreted as being “not externally regulated” (see chapter 3.1). The factor loadings for the second component are high for human capital, reputation, and personal requirement but low for fun seeking, peer recognition, and pro-social motivation. Human capital, reputation, and personal requirement are all personal investments related to the spending of scare resources in the hope for a future return with a certain interest rate. Therefore, this component was interpreted as personal investment. Pro-social motivation is also an investment. However, it is not geared toward the individual but toward the well-being of others. Pro-social motivation is therefore an investment for society. That pro-social motivation loads negatively on the second component fits this interpretation perfect.

Rotated Component Matrix[ab]		
Motivation	**Not external regulation**	**Personal investment**
Fun seeking	**0.748**	0.279
Pro social	**0.715**	-0.175
Peer recognition	**0.640**	0.325
Personal requirements	0.232	**0.631**
Reputation	-0.151	**0.821**
Human capital	0.322	**0.469**

N=177
[a] Components extracted with principal component analysis
[b] Varimax rotated

Figure 169: Factor analysis of motivation types: Rotated component matrix

The results of the factor analysis discussed above and the corresponding component plot in rotated space (see Figure 170) support hypothesis 4. The motivation types fun seeking, pro-social motivation, and peer recognition cluster together on the one hand and the motivation types personal requirement, personal requirement, investment in reputation, and investment in human capital cluster together on the other hand.

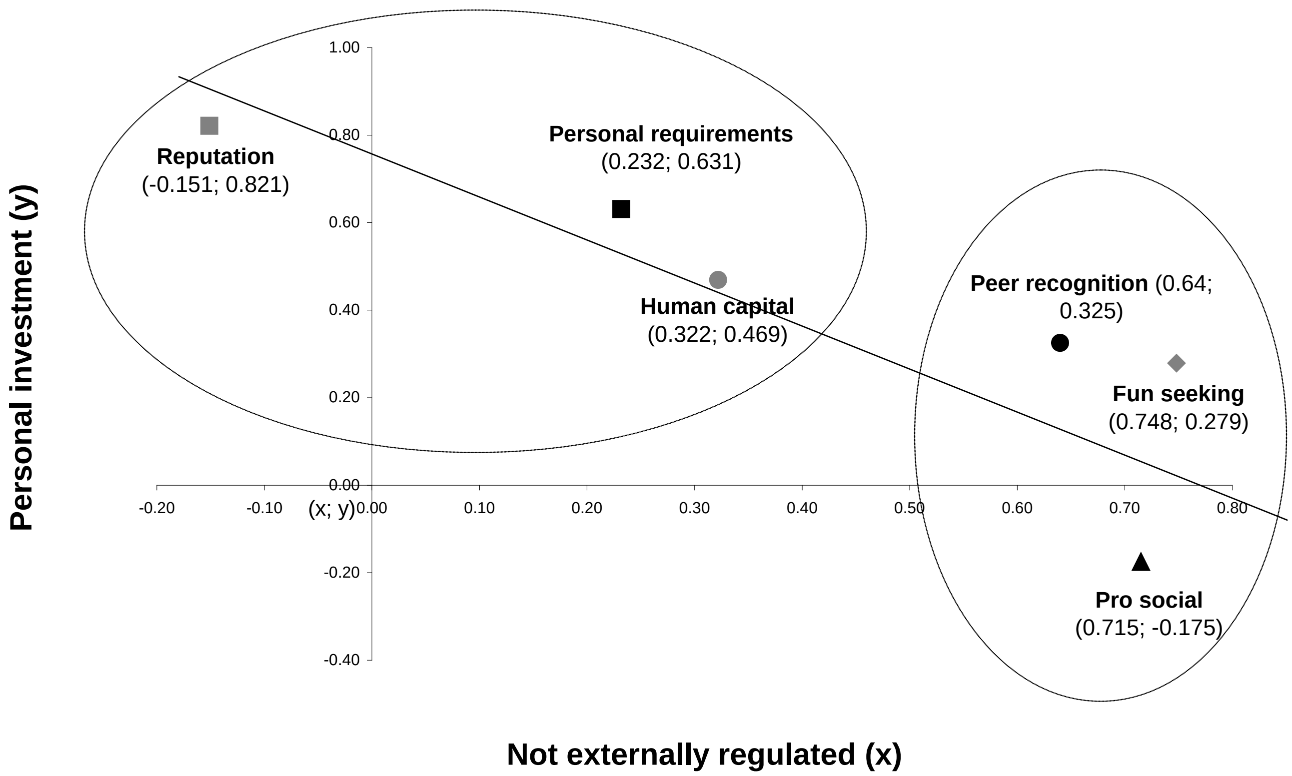


Figure 170: Factor analysis of motivation types: Component plot in rotated space

11.4.2 Motivation and governance structures respectively mechanisms

11.4.2.1 Assignment for project positions

Hypothesis 5

Hypothesis 5 suggested that the fraction of projects with no special mechanisms to assign project positions is considerably lower for very small projects with few programmers than for bigger projects. Figure 171 shows that the fraction of projects that are too small for having mechanisms to deal with the assignment of special positions decreases with the number of programmers. For very small projects roughly three-quarters of the projects are too small for having such mechanisms. For projects with 5 to 9 programmers half of the projects are too small, for medium projects with 10 to 25 programmers roughly one-seventh, and for big projects with more than 25 programmers there are no projects that are too small for having such mechanisms. The difference between very small and all other projects is significant. [682] Hypothesis 5 is therefore supported.

Governance mechanism and number of programmers								
	Number of programmers							
Governance mechanisms	*Very small project 1-4*		*Small project 5-9*		*Medium project 10-25*		*Big project >25*	
Not too small	27.1%	27.6%	50.0%	52.0%	84.4%	84.4%	92.9%	100.0%
Too small	71.2%	72.4%	46.2%	48.0%	15.6%	15.6%	0.0%	0.0%
Missing	1.7%	-	3.8%	-	0.0%	-	7.1%	-
N	59	58	26	25	32	32	28	26

Figure 171: Governance mechanisms and the number of programmers

Hypotheses 6 to 8

Hypotheses 6 to 8 suggested that motivation is related to the assignment of project positions. It was hypothesized that fun seeking is positively related to the assignment by informal consent, that peer recognition is negatively related to the assignment by appointment, and that reputation investment is positively related to the assignment by elections. In fact, the following empirical results were found (see Figure 172):

- As can be seen in Figure 173a fun seeking is lower in projects that assign project positions by informal consent. Therefore, hypothesis 6 is clearly not supported.
- The motivation related to peer recognition is significantly higher in projects in which special positions are appointed hierarchically (Figure 173b). This result is completely in contrast to hypothesis 7 that is therefore not supported.

[682] Chi-square test with 1 df, P = 0.000001.

- As assumed in hypothesis 8 the motivation related to investment in reputation is higher in projects in which there are elections (Figure 173c). However, this result is not significant (see 90% confidence intervals). Hypothesis 8 is therefore not supported.

Motivation	Decision making			
	Too small	Informal consent	Appointment	Election
Fun seeking	4.13	4.09	**4.24**	4.17
Pro-social	3.94	**3.95**	**3.95**	3.62
Peer recognition	3.18	3.41	**3.82**	3.57
Personal requirement	3.81	3.42	**3.65**	3.73
Reputation	3.02	3.17	3.18	**3.26**
Human capital	4.20	**4.15**	3.93	**4.16**

Figure 172: Motivation and the assignment of special positions

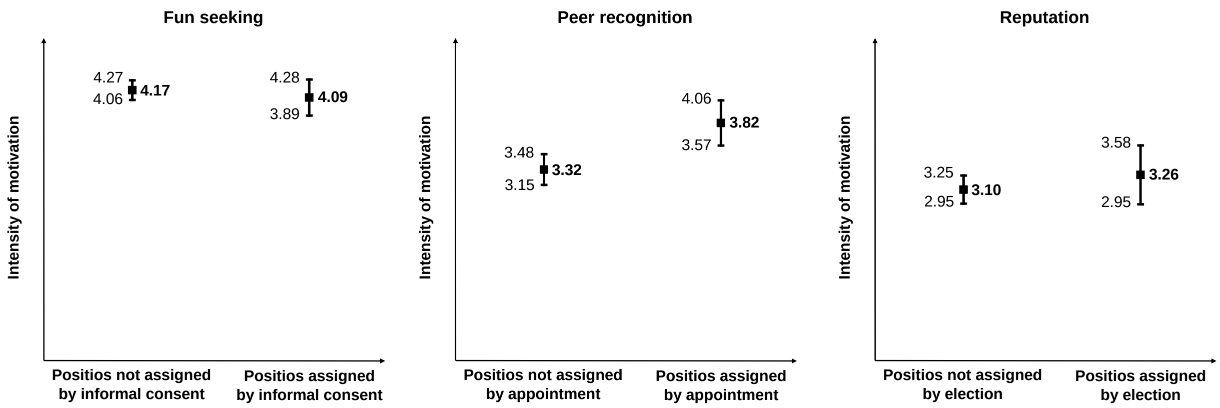


Figure 173abc: Motivation and the assignment of special positions: 90% confidence interval

As discussed above the relationship between the assignment of special positions and motivation is not according to the hypotheses. In fact, the following relationships are observed (see Figure 172):

- Fun seeking, peer recognition, and personal requirement are the highest if special project positions are appointed hierarchically.
- Pro-social motivation is the highest in projects that either assign special positions by informal consent or that appoint them hierarchically.
- The motivation related to human capital is the highest in projects that conduct elections and in projects that assign positions by informal consent.
- The motivation related to investment of reputation is the highest in projects that conduct elections.

11.4.2.2 Structures and mechanisms

Hypotheses 9 and 10

It was suggested in hypothesis 9 that fun seeking is positively related to autonomy, competence, relatedness, and positive feedback. It was also assumed in hypothesis 10 that fun seeking is positively related to projects that would be missed by developers if they were to disappear.

Fun seeking is, as suggested, significantly and positively correlated to competence, positive feedback, and to the degree a project would be missed by the developers if it were to disappear (see Figure 174 and Figure 174). However, autonomy and relatedness do not seem to be related to fun seeking. Therefore, hypotheses 9bd and 10 are supported while hypotheses 9ac are not supported.

Structures and mechanisms: Fun seeking						
		Autonomy	**Competence**	**Relatedness**	**Positive feedback**	**Missed by developers**
Fun seeking	Correlation[a]	0.06	0.17	0.07	0.13	0.18
	P	0.1593	0.0039	0.1407	0.0193	0.0037
	N	159	157	161	161	159

[a] Kendall-Tau-b correlation, one-sided

Figure 174: Structures and mechanisms: Fun seeking

Hypotheses 11 to 13

It was suggested in hypothesis 11 that pro-social motivation is positively related to happiness with the license scheme of a project. It was also assumed in hypothesis 12 that the adoption of a strong copyleft license is positively associated with pro-social motivation. Hypothesis 13 suggested that pro-social motivation is positively related to projects that would be missed by users and the entire community if it were to disappear.

The two hypotheses related to licenses are supported by the data (see Figure 175a). Pro-social motivation is significantly higher in projects with strong copyleft licenses and if subjects are happy with the license scheme. Hypotheses 11 and 12 are therefore supported. However, there is no significant relationship between pro-social motivation and the degree to which projects would be missed by the community or the users if they were to disappear (see Figure 175b). Hypotheses 13ab are not supported.

Structures and mechanisms: Pro social motivation			
		Missed by community	**Missed by users**
Pro social motivation	Correlation[a]	-0.01	0.05
	P	0.4683	0.2044
	N	149	154

[a] Kendall-Tau-b correlation, one-sided

Structures and mechanisms: Pro social motivation					
		Happy with license		**Strong copyleft license**	
Pro social motivation	Degree of freedom[a]	1		1	
	P	0.0021		0.0200	
	N	131[b]	24[c]	110[d]	25[e]

[a] Kruskal-Wallis one-way analysis of variance
[b] Happy with license; [c] Not happy with license
[d] Strong copyleft license; [e] No strong copyleft license

Figure 175ab: Structures and mechanisms: Pro-social motivation

Hypotheses 14 to 15

It was suggested in hypotheses 14 and 15 that peer recognition is positively related to relatedness, to the number of programmers, and to the degree a project would be missed by its developers or the entire community if it were to disappear.

Only relatedness is significantly correlated to peer recognition while the number of programmers, the degree a project is missed by developers, and the degree a project is missed by the community are not (see Figure 176). Hypothesis 14a is supported while hypotheses 14b and 15ab are not supported.

Structures and mechanisms: Peer recognition					
		Relatedness	**Missed by developers**	**Missed by community**	**Number of programmers**
Peer recognition	Correlation[a]	0.14	0.10	-0.03	0.07
	P	0.0431	0.1215	0.3734	0.1877
	N	96	93	90	86

[a] Kendall-Tau-b correlation, one-sided

Figure 176: Structures and mechanisms: Peer recognition

Hypotheses 16 to 17

Personal requirement is thought to be positively related to projects that would be missed by the users if they were to disappear and to projects that are included in a major Linux distribution (hypotheses 16 and 17).

The degree a project is missed by its users is significantly and positively correlated to personal requirement (see Figure 177a). Hypothesis 16 is therefore supported. Projects that are included in major Linux distributions are not positively related to personal requirement (see Figure 177b). Hypothesis 17 is not supported.

Structures and mechanisms: Personal requirement		
		Missed by users
Personal requirement	Correlation[a]	0.15
	P	0.0120
	N	154

[a] Kendall-Tau-b correlation, one-sided

Structures and mechanisms: Personal requirment			
		Included in distribution	
Personal requirement	Degree of freedom[a]	1	
	P	0.2476	
	N	66[b]	74[c]

[a] Kruskal-Wallis one-way analysis of variance
[b] Included in distribution; [c] Not included in distribution

Figure 177ab: Structures and mechanisms: Personal requirement

Hypotheses 18 to 21

In hypotheses 18 to 21 it was assumed that investment in reputation is positively related

- to the degree a project would by missed by the entire community and the users if it were disappear,
- to the number of different means to credit contributions,
- to the involvement of commercial firms, and
- to the number of developers in a project.

In Figure 178 it can be recognized that all the above mentioned items are positively and significantly correlated to reputation investment. Hypotheses 18ab, 19b, 20, and 21 are therefore clearly supported.[683]

Structures and mechanisms: Reputation investment							
		Missed by community	Missed by users	Number of modes to credit	Company contribution beginning	Company contribution right now	Number of programmers
Reputation investment	Correlation[a]	0.22	0.19	0.14	0.11	0.31	0.13
	P	0.0003	0.0020	0.0170	0.0365	0.0000	0.0170
	N	148	153	162	157	157	136

[a] Kendall-Tau-b correlation, one-sided

Figure 178: Structures and mechanisms: Reputation investment

In hypothesis 19a it was assumed that the existence of credit mechanisms is positively related to reputation investment. The corresponding motivation is indeed higher in projects that have at least one mechanism to credit. However, the relationship is clearly not significant.[684] Hypothesis 19a is therefore not supported. A reason for this result may be that there are almost no projects (only 7 projects) without any mechanism to credit (see Figure 130) that makes the corresponding statistical analysis unreliable.

[683] No other type of motivation is significantly related to the degree companies are currently involved in development.
[684] Kruskal-Wallis one-way analysis of variance, df $=1$, $P=0.543$.

Hypotheses 22 to 24

According to hypotheses 22 to 24 investment in human capital should be positively related to the degree decisions are explained, to feedback giving, to competence, and to the degree a project would be missed by developers if it were to disappear.

The data supports the assumptions that the motivation related to human capital is positively correlated to feedback giving, the degree a project is missed by developers, and competence (see Figure 179). However, investment in human capital is not related to the degree decisions are explained. Hypotheses 22a, 23, and 24 are therefore supported while hypothesis 22b is not supported.

Structures and mechanisms: Human capital					
		Decision explained	**Feedback**	**Missed by developers**	**Competence**
Human capital	Correlation[a]	0.00	0.31	0.16	0.18
	P	0.4835	0.0000	0.0060	0.0018
	N	154	157	155	154

[a] Kendall-Tau-b correlation, one-sided

Figure 179: Structures and mechanisms: Investment in human capital

11.4.2.3 Paid developers and the involvement of firms

Hypotheses 25 to 28

It was suggested that paid developers are more likely engaged in projects to which commercial firms contribute as well (hypothesis 25) and in projects in a later development phase (hypothesis 28). Hypothesis 26 assumed that fun seeking is negatively related and that personal requirement, investment in human capital, and investment in reputation are not related to the status of being paid. It was further assumed in hypothesis 27 that the development phase is positively related to the involvement of commercial firms.

As can be seen in Figure 180 paid developers contribute significantly more to projects to which companies contribute as well. Hypothesis 25 is therefore supported.

Contrary to hypothesis 26a there is no significant negative relationship between the status of being paid and fun seeking. Hypothesis 26a is therefore not supported. As suggested there is no significant relationship between investment in human capital respectively personal requirement and the status of being paid. However, there is a significant relationship between investment in reputation and the status of being paid. Paid developers are more motivated by investment in reputation. Hypothesis 26b is therefore not supported.

As suggested by hypothesis 28 paid developers are more likely engaged in projects in a later development phase than non-paid developers are. Hypothesis 28 is supported.

		Paid developers					
		Company contribution right now	Fun seeking	Personal requirement	Reputation	Human capital	Development phase
Paid developers	Degree of freedom[a]	1	1	1	1	1	1
	P	0.0053	0.2324	0.5636	0.0165	0.9455	0.0077
	N	141[b] 14[c]	146[b] 14[c]	140[b] 14[c]	141[b] 13[c]	143[b] 13[c]	136[b] 14[c]

[a] Kruskal-Wallis one-way analysis of variance
[b] Not paid; [c] Paid

Figure 180: Paid developers and motivation

In hypothesis 27 it was suggested that the development phase is positively related to the involvement of commercial firms. Indeed, the involvement of commercial firm is significantly higher in later development phases.[685] Further support for hypothesis 27 has already been presented in chapter 11.2.2 (company contributions at the beginning and recently). Consequently, hypothesis 27 is supported.

11.4.3 Interaction of motivation and governances structures respectively mechanisms

Hypothesis 29

In hypothesis 29 it was suggested that after having controlled for demographic variables, motivation, and governance structures respectively mechanisms interaction effects between motivation and governance structures respectively mechanisms should help to explain satisfaction with FOSS involvement.

To test hypothesis 29 hierarchical regressions are run (see Jaccard & Turrisi 2003: 11-12). In a first step demographic variables (e.g. age) are introduced in model 1. Afterwards, the motivation in question (e.g. fun seeking) is introduced in model 2. In model 3 governance structures and mechanisms in question (e.g. autonomy and positive feedback) are added. Finally, in model 4 the interaction items are introduced. If hypothesis 29 is true the interaction items should increase the explained variance significantly.

Statistical analysis shows that for no motivational type the suggested interaction effects increase explained variance significantly (see Appendix III and Figure 181). For example for fun seeking the interaction items explain only 7% (see Figure 181). This increase as well as the increase for pro-social motivation (8%), for peer recognition (10%), for personal requirement (1%), for human capital (1%), and for reputation investment (1%) are not significant. Hypothesis 29 is therefore not supported.

[685] Kendall-Tau-b correlation = 0.11, one-sided, P = 0.05, N = 156.

Satisfaction with FOSS particpation: Fun seeking						
Model	**R^2**	**Change in R^2**	**Change in F**	**df1[a]**	**df2[b]**	**Significance in change of F**
1 Demographics	0.055	0.055	0.814	6	84	0.562
2 Demographics, Motivation	0.055	0.000	0.007	1	83	0.935
3 Demographics, motivation, structures and mechanisms	0.186	0.131	2.512	5	78	0.037
4 Demographics, motivation, structures and mechanisms, interaction effects	0.256	0.070	1.363	5	73	0.248

[a] Nominator degress of freedom
[b] Denominator degress of freedom

Figure 181: Interaction effects and satisfaction with FOSS development: Fun seeking[686]

11.4.4 Motivation and type of engagement

Hypotheses 30 to 35

In hypotheses 30 to 35 it was suggested that the type of engagement in FOSS development is related to motivation:

- Hypothesis 30 suggested that fun seeking is positively related to adding/developing features and to reviewing source code. Indeed, the motivation related to fun seeking is higher for subjects who add or develop features[687] and who review source code.[688] Hypotheses 30ab are therefore supported.
- Hypothesis 31 suggested that pro-social motivation is positively related to translating, marketing/promoting, creating graphics, writing how FAQ's, and donating. Subjects who engage in translating[689], creating graphics[690], writing FAQ's[691], and donating[692] have no higher pro-social motivation. However, subjects who engage in marketing/promoting have a higher pro-social motivation.[693] Only hypothesis 31b but not hypotheses 31acde are supported.

[686] Demographic variables included are gross income, student (yes/no), education, working hours, programming experience, and age. The included motivation item is the fun seeking scale. Items related to structures and mechanisms are feedback, missed by developers, autonomy, competence, and relatedness. The interaction items are fun_x_feedback, fun_x_missed by developers, fun_x_autonomy, fun_x_competence, and fun_x_relatedness.
For more data see Figure 204.
[687] Kruskal-Wallis one-way analysis of variance, df = 1, P = 0.042.
[688] Kruskal-Wallis one-way analysis of variance, df = 1, P = 0.092.
[689] Kruskal-Wallis one-way analysis of variance, df = 1, P = 0.937.
[690] Kruskal-Wallis one-way analysis of variance, df = 1, P = 0.048 (wrong direction, pro-social motivation is lower).
[691] Kruskal-Wallis one-way analysis of variance, df = 1, P = 0.181.
[692] Kruskal-Wallis one-way analysis of variance, df = 1, P = 0.818.
[693] Kruskal-Wallis one-way analysis of variance, df = 1, P = 0.072.

- Hypothesis 32 suggested that peer recognition is positively related to reviewing source code and helping users. The motivation related to peer recognition is neither significantly higher for subjects who review code[694] nor for subjects who help users.[695] Hypothesis 32 is not supported.
- Hypothesis 33 suggested that personal requirement is positively related to reporting bugs, fixing bugs, and adding/developing features. The motivation related to personal requirements is significantly higher for subjects who add/develop features[696] but not for subjects who fix[697] or report bugs.[698] Hypothesis 33c is supported but not hypothesis 33ab.
- Hypothesis 34 suggested that reputation investment is positively related to adding/developing features and to promoting/marketing. The motivation related to reputation investment is not significantly higher for subjects who add or develop features.[699] Hypothesis 34a is not supported. However, reputation investment is significantly higher for subjects who engage in promoting and marketing.[700] Hypothesis 34b is therefore supported.
- Hypothesis 35 suggested that investment in human capital is positively related to adding/developing features, bug fixing, and reviewing source code. The motivation related to investment in human capital is significantly higher for subjects who review source code[701] but not for subjects who fix bugs[702]. For subjects who add/develop features[703] the relationship is almost significant. Hypothesis 35c is therefore supported but not hypotheses 35ab.

No relationship between pro-social motivation and mundane tasks such as translating, creating graphics, writing FAQ's, and donating could be found. The question arises: If not pro-socially motivated individuals who else engage in mundane tasks? There are no significant relationships. However, arguable empirical results suggest that translating and writing FAQ's are positively related to investment in human capital, that creating graphics is positively related to personal requirements, and that donating is positively related to peer recognition.

11.4.5 Motivation and community affiliation

Hypothesis 36

It was suggested in hypothesis 36 that pro-social motivation is positively related to being part of the free software and negatively related to being part of the open source community. As can be seen in Figure 182d pro-social motivation is significantly higher for subjects who consider being part of the free software community (see 90% confidence intervals). Being part of the free software community has no significant

[694] Kruskal-Wallis one-way analysis of variance, df $=1$, $P=0.145$.
[695] Kruskal-Wallis one-way analysis of variance, df $=1$, $P=0.837$.
[696] Kruskal-Wallis one-way analysis of variance, df $=1$, $P=0.069$.
[697] Kruskal-Wallis one-way analysis of variance, df $=1$, $P=0.154$.
[698] Kruskal-Wallis one-way analysis of variance, df $=1$, $P=0.530$.
[699] Kruskal-Wallis one-way analysis of variance, df $=1$, $P=0.362$.
[700] Kruskal-Wallis one-way analysis of variance, df $=1$, $P=0.032$.
[701] Kruskal-Wallis one-way analysis of variance, df $=1$, $P=0.068$.
[702] Kruskal-Wallis one-way analysis of variance, df $=1$, $P=0.352$.
[703] Kruskal-Wallis one-way analysis of variance, df $=1$, $P=0.101$.

influence on fun seeking, investment in reputation, investment in human capital, peer recognition, and personal requirement (see Figure 182abcef). Hypothesis 36 is therefore supported.

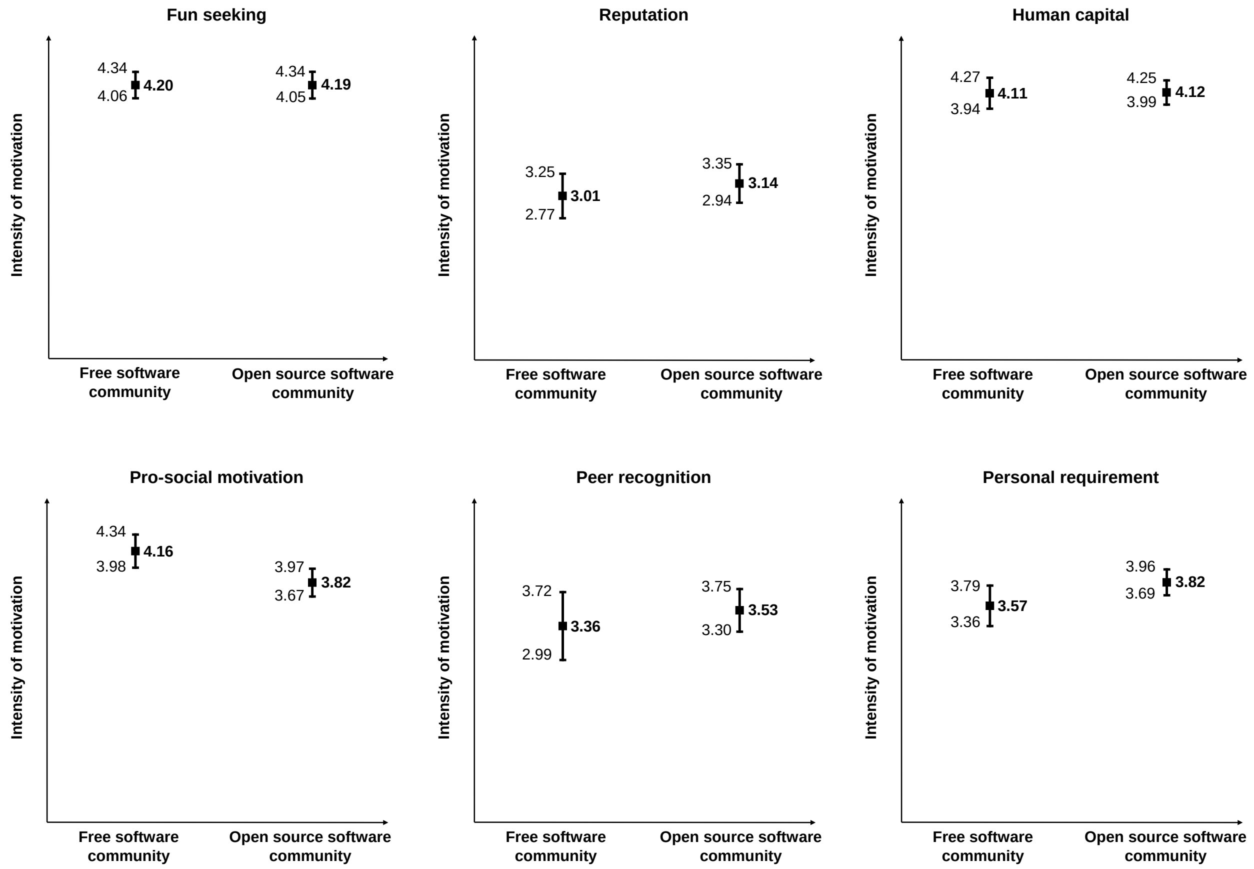


Figure 182abcdef: 90% confidence interval of community - motivation interaction

Hypothesis 37

It was suggested in hypothesis 37 that being part of the free software community is positively related to projects with a strong copyleft license.

From all subjects who consider being part of the free software community 89% contribute to a project with a strong copyleft license while 11% contribute to a project with another license (see Figure 183). In the case of subjects who consider being part of the open source community 66% contribute to a project that has adapted a strong copyleft license and 34% to a project with another license. This difference between the subjects who consider being part of the free or the open source community is significant.[704] Hypothesis 37 is supported.

[704] Chi-square test with 1 df, P = 0.006.

<table>
<tr><th colspan="9">Classified FOSS licenses and community affiliation</th></tr>
<tr><th></th><th>Free Software</th><th>Open Source</th><th>Free Software</th><th>Open Source</th><th>Free Software</th><th>Open Source</th><th>Free Software</th><th>Open Source</th></tr>
<tr><td>Strongly copylefted</td><td>85.42%</td><td>63.64%</td><td>89.13%</td><td>66.04%</td><td rowspan="2">95.65%</td><td rowspan="2">81.13%</td><td rowspan="3">95.83%</td><td rowspan="3">96.36%</td></tr>
<tr><td>Weakly coplefted</td><td>6.25%</td><td>14.55%</td><td>6.52%</td><td>15.09%</td></tr>
<tr><td>Not copylefted</td><td>4.17%</td><td>18.18%</td><td>4.35%</td><td>18.87%</td><td>4.35%</td><td>18.87%</td></tr>
<tr><td>Not decided</td><td>4.17%</td><td>3.64%</td><td>-</td><td>-</td><td>-</td><td>-</td><td>4.17%</td><td>3.64%</td></tr>
<tr><td>N</td><td>48</td><td>55</td><td>46</td><td>53</td><td>46</td><td>53</td><td>48</td><td>55</td></tr>
</table>

Figure 183: Licenses and community affiliation

Roughly 96% of the members of the free software community contribute to a project with a copyleft license (either strong or weak) while only 81% of the members of the open source community do so. This difference is significant as well.[705]

Hypothesis 38

In hypothesis 38 it was suggested that being part of the free software community is negatively related to business interests.

The fraction of subjects with a business interest is significantly lower for members of the free software community (see Figure 184).[706] Only 21% of the members of the free software community have a business interest that is related to their engagement. For the open source community almost half of the members have a business relationship. Hypothesis 38 is supported.

Business relationship			
	Business relationshp	**No business relationship**	**N**
Free Software community	21.15%	78.85%	52
Open Source community	43.64%	56.36%	55

Figure 184: Community affiliation and business interests

Interestingly, the engagement of the members of the free software community is significantly and negatively related to the contribution of commercial firms (see Figure 185).[707] The fraction of subjects of the free software community who strongly disagree that companies made important contributions is with 47% by far higher than the responding fraction in the open source software community (23%). This finding is reproduced by measures of central tendency. The median of the free software community is 2, the one of the open source

[705] Chi-square test with 1 *df*, P = 0.026.
[706] Chi-square test with 1 df, P = 0.005
[707] Chi-square test with 4 *df*, P = 0.005

community 3. It can therefore be concluded that a subject who belongs to the open source community does more likely add to a project to which companies make important contributions.

Commercial firms made important contributions recently						
	Strongly disagree1	2	3	4	Strongly agree 5	N
Free Software community	47.06	7.84	19.61	21.57	3.92	51
Open Source community	22.81	15.79	29.82	22.81	8.77	57

Figure 185: Community affiliation and the involvement of commercial firms

There is also a statistical significant difference between the two different communities concerning gross income.[708] Subjects considering being part of the free software have a lower gross income than the ones considering being part of the open source community. The median income of a member of the free software community is between 501 and 2000 US$ while the same value for the open source community is between 2001 and 4000$. This difference cannot be attributed to age or the status of being a student. Concerning these two items there is no statistical significant difference between the two communities.

Gross income					
	< 500 US$	501 - 2000 US$	2001 - 4000 US$	> 4001 US$	N
Free Software community	32.43	21.62	35.14	10.81	37
Open Source community	15.38	23.08	38.46	23.08	39

Figure 186: Gross income and community affiliation[709]

Although no hypothesis is connected to development phases and community affiliation an interesting result is briefly discussed below. Based on a Goodness of fit test it can be assumed that there is a significant difference between the two communities as far as their participation in FOSS projects in different phases is concerned.[710] The data suggests that there are different pattern for the two communities (see Figure 187):

- For the open source community the development phase of a project has no influence for participation. There are in all four phases about one fourth of the subjects belonging to the open source community. This observation is backed up by statistical analysis that shows no significant difference between observed and expected values under the assumption of uniform distribution.[711]

[708] Chi-square test with 3 *df*, P = 0.025.

[709] The categories “4001-7000” and “ > 7001” had to been merged because otherwise there would have been cells with size below 5. Because chi-square tests are based on nominal scaled variables this operation is valid.

[710] Chi-square test with 3 *df*, P = 0.005.

[711] Chi-square test with 3 *df*, P = 0.978.

- There is a major influence of the development phase on the participation of free software community members. Contribution of subjects increases during project development. This increase is almost significant.[712] This directs to the implication that the contribution of free software community member is positively related to the development phase of a project.

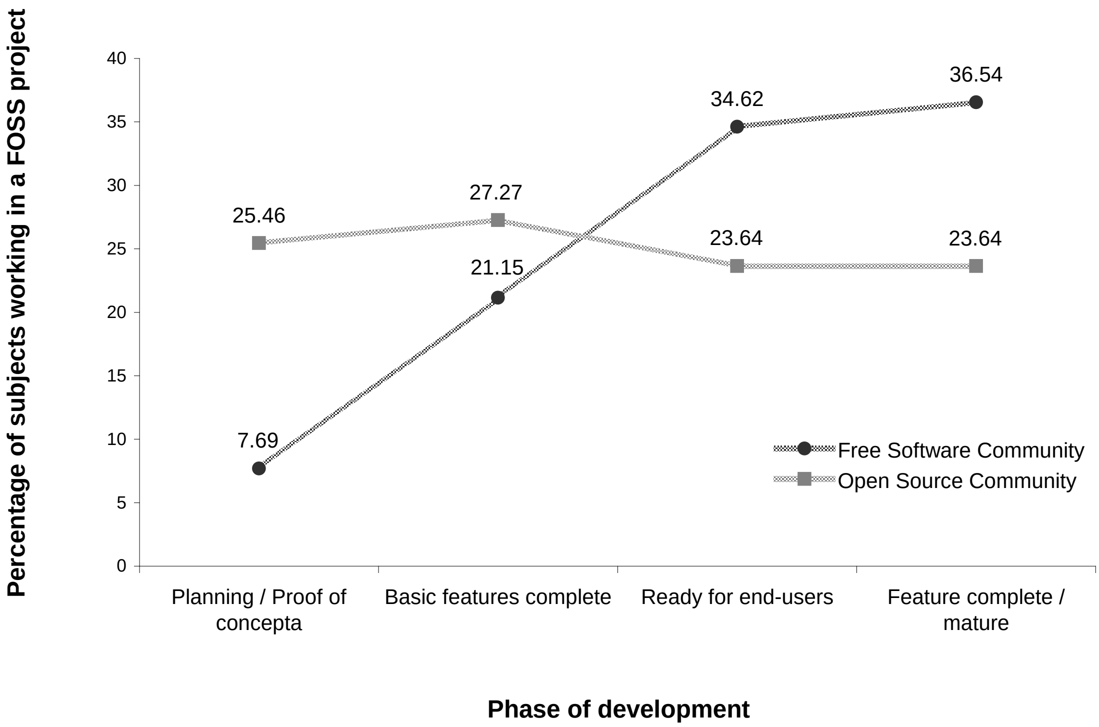
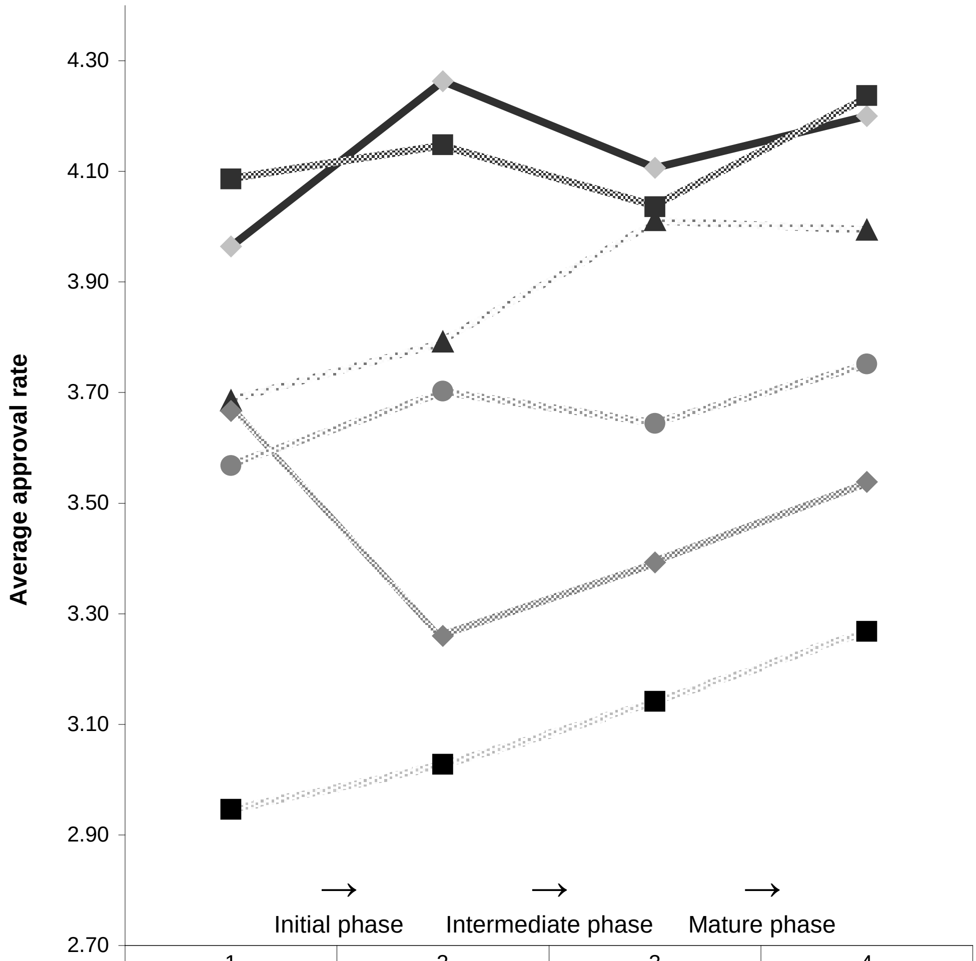


Figure 187: Percentage of subjects in different development phases

11.4.6 Motivation and development phases

In the following it is discussed how motivation changes contingent on the development phases “initial phase”, “intermediate phase”, and “mature phase”. As can be seen in Figure 188 the composition of motivation seems to change during FOSS development.

[712] Chi-square test with 3 *df*, P = 0.105.



	1	2	3	4
Fun seeking	3.96	4.26	4.11	4.20
Pro social	3.69	3.79	4.01	3.99
Peer recognition	3.67	3.26	3.39	3.54
Personal requirements	3.57	3.70	3.64	3.75
Reputation	2.95	3.03	3.14	3.27
Human capital	4.09	4.15	4.04	4.24

Phase of development

Figure 188: Motivation and development phases: Chart of results

In the following it is discussed how each type of motivation changes contingent on the development phase. It should be noted that in the line charts presented below the scales of the y-axis are heavily adjusted so that differences are highly visible. Thereby even small differences look like being big. For example autonomy (see Figure 199) increases from 3.98 to 3.99, to 4.03, and finally decreases to 3.98. Therefore, autonomy remains practically constant in all phases. In contrast, the line chart looks like that there is a huge change in the intermediate phase. To make such effects transparent the values have always been inserted in the line charts. For convenience, the lines in the charts are additionally smoothed.

Curve sketching fun seeking

In Figure 189b one can see the prediction and in Figure 189a the actual data related to fun seeking in different development phases. It was suggested that motivation decreases during development. However, motivation increases, decreases, and increases. Out of three predictions one was correct. The actual distribution of fun seeking in different development phases does not look at all similar to the predictions.

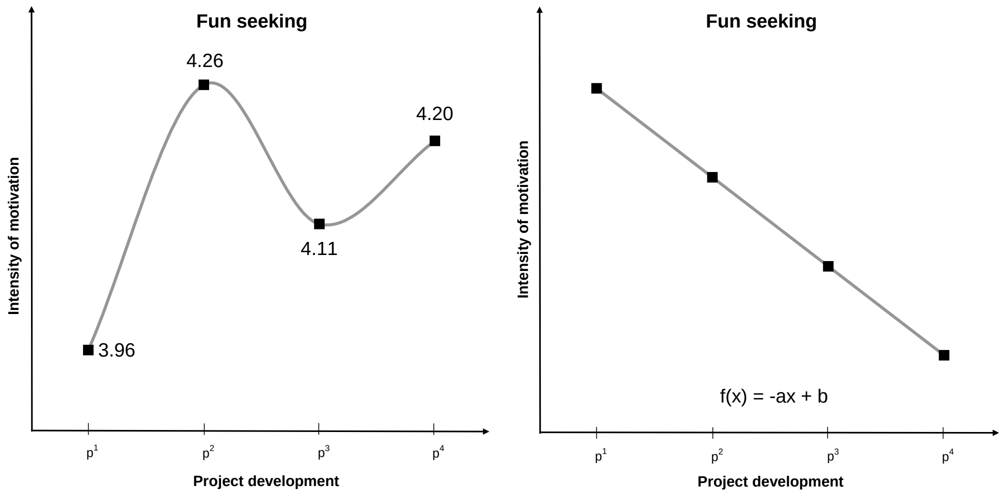


Figure 189ab: Fun seeking and development phases: Results and predictions

<u>Summary: Fun seeking and development phases</u>

Phase	*Prediction*	*Data*	*Correspondence prediction/data*
Initial phase	Motivation ***decreases***	Motivation ***increases*** from 3.96 to 4.26.	No
Intermediate phase	Motivation ***decreases***	Motivation ***decreases*** from 4.26 to 4.11.	Yes
Mature phase	Motivation ***decreases***	Motivation ***increases*** from 4.11 to 4.20.	No

Curve sketching pro-social motivation

In Figure 190b one can see the prediction and in Figure 190a the actual data related to pro-social motivation in different development phases. It was suggested that motivation increase, remain similar, and decreases. In fact, motivation increases, increases again, and decreases. Out of three predictions two were correct. The decrease in motivation in the mature phase is very small. The predicted n-shaped distribution does therefore not seem to be correct. The distribution of pro-social motivation in the different development phases resembles a logarithmic function discussed for investment in reputation.

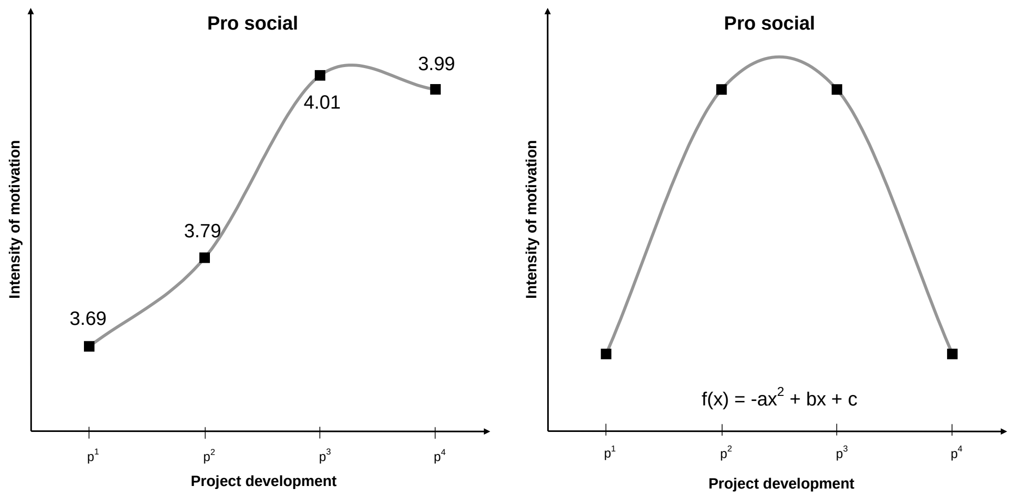


Figure 190ab: Pro-social motivation and development phases: Results and predictions

Summary: Pro-social motivation and development phases

Phase	*Prediction*	*Data*	*Correspondence prediction/data*
Initial phase	Motivation ***increases***	Motivation ***increases*** from 3.69 to 3.79.	Yes
Intermediate phase	Motivation ***remains similar***	Motivation ***increases*** from 3.79 to 4.01.	No
Mature phase	Motivation ***decreases***	Motivation ***decreases*** slightly from 4.01 to 3.99.	Yes

Curve sketching peer recognition

In Figure 191b one can see the prediction and in Figure 191a the actual data related to peer recognition in different development phases. It was suggested that motivation decreases, remain similar, and finally increases. The data supports this pattern and the data fits the predicted distribution very good. Out of three predictions two were correct.

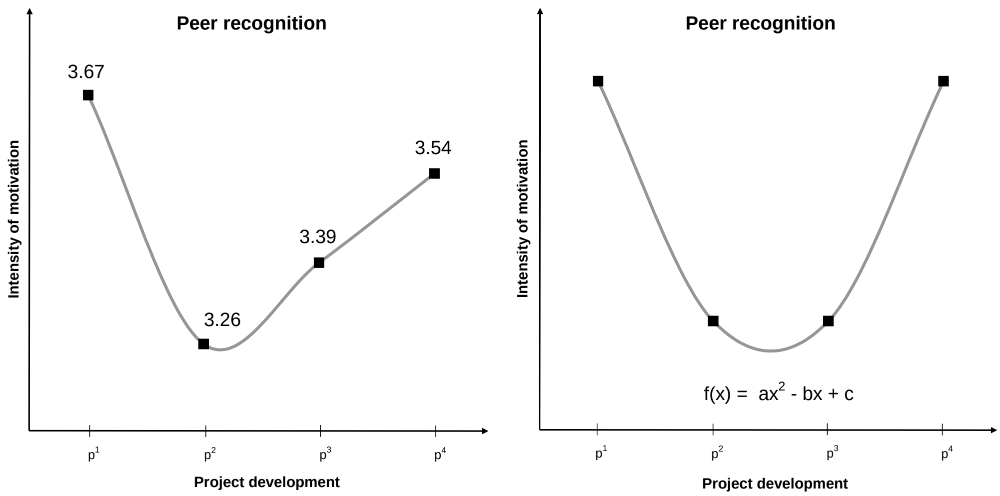


Figure 191ab: Peer recognition and development phases: Results and predictions

Summary: Peer recognition and development phases

Phase	*Prediction*	*Data*	*Correspondence prediction/data*
Initial phase	Motivation ***decreases***	Motivation ***decreases*** from 3.67 to 3.26.	Yes
Intermediate phase	Motivation remains similar	Motivation ***increases*** from 3.25 to 3.39.	No
Mature phase	Motivation ***increases***	Motivation ***increases*** from 3.39 to 3.54.	Yes

Curve sketching personal requirements

In Figure 192b one can see the prediction and in Figure 192a the actual data related to personal requirements in different development phases. It was assumed that motivation increases constantly during development. In fact, motivation increases, decreases, and increases. Out of three predictions two were correct. The actual distribution seems to fit the predicted distribution to some extent.

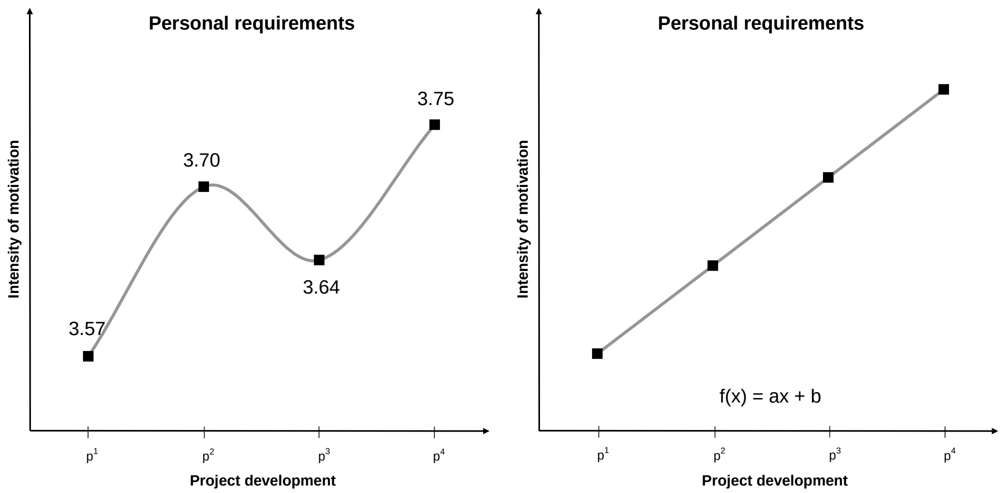


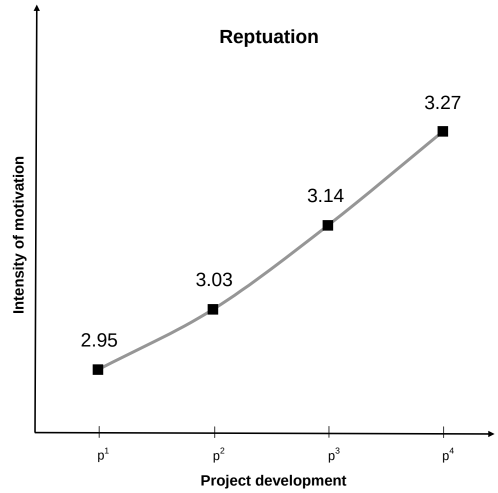
Figure 192ab: Personal requirement and development phases: Results and predictions

Summary: Personal requirement and development phases

Phase	*Prediction*	*Data*	*Correspondence prediction/data*
Initial phase	Motivation ***increases***	Motivation ***increases*** from 3.57 to 3.70.	Yes
Intermediate phase	Motivation ***increases***	Motivation ***decreases*** from 3.70 to 3.64	No
Mature phase	Motivation ***increases***	Motivation ***increases*** from 3.64 to 3.75.	Yes

Curve sketching investment in reputation

In Figure 193b one can see the prediction and in Figure 193a the actual data related to investment in reputation in different development phases. It was suggested that motivation increases always during development that proved to be correct. Therefore, out of three predictions all were correct. However, the predicted pattern is only partly supported. That motivation flattens in the mature phase as suggested seems not be supported. Reputation seems to increase linearly with development going on.



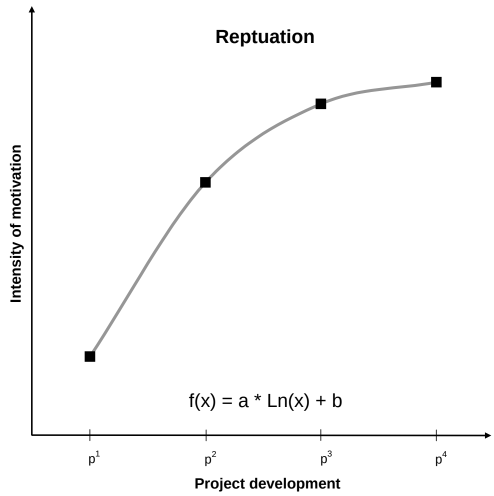


Figure 193ab: Reputation investment and development phases: Results and predictions

Summary: Reputation investment and development phases

Phase	*Prediction*	*Data*	*Correspondence prediction/data*
Initial phase	Motivation ***increases***	Motivation ***increases*** from 2.95 to 3.03.	Yes
Intermediate phase	Motivation ***increases***	Motivation ***increases*** from 3.03 to 3.14	Yes
Mature phase	Motivation ***increases***	Motivation ***increases*** from 3.14 to 3.27	Yes

Curve sketching human capital

In Figure 194b one can see the prediction concerning the change of motivation related to human capital in different development phases. It was suggested that motivation increases, decreases, and increases during development. In Figure 194a the empirical results are presented. Motivation increases, decreases, and increases again. Out of three predictions all were correct. However, for two predictions the change in motivation is not very big. In general, the predicted distribution and the actual distribution look very similar.

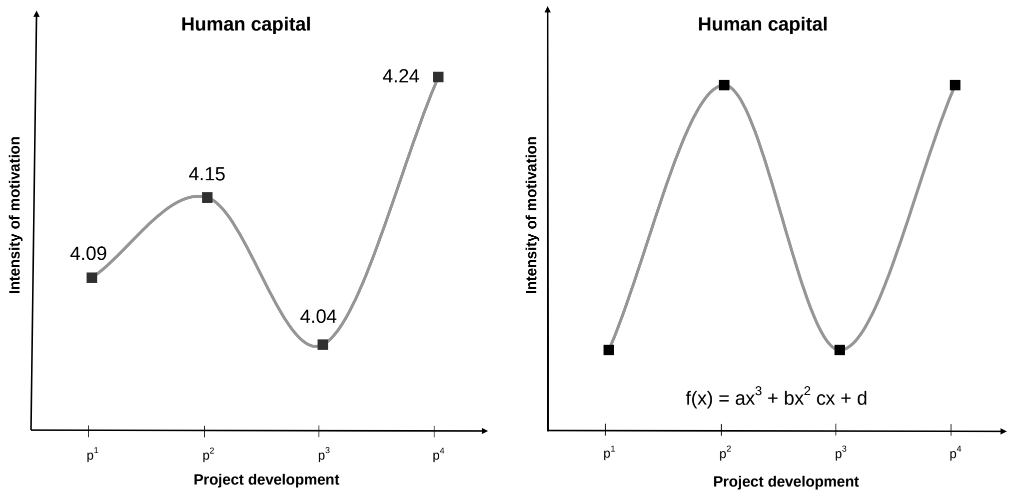


Figure 194ab: Human capital and development phases: Results and predictions

Summary: Human capital and development phases

Phase	*Prediction*	*Data*	*Correspondence prediction/data*
Initial phase	Motivation ***increases***	Motivation ***increases*** slightly from 4.09 to 4.15.	Yes
Intermediate phase	Motivation ***decreases***	Motivation ***decreases*** from 4.15 to 4.04.	Yes
Mature phase	Motivation ***increases***	Motivation ***increases*** sharply from 4.04 to 4.24.	Yes

Curve sketching: A short overview over the results

It can be concluded that the assumptions concerning motivation and development phases are:

- very accurate for human capital and peer recognition,
- quite accurate for reputation,
- middling for pro-social motivation and personal requirement, and
- very poor for fun seeking.

Hypothesis 39a

In chapter 8.2.6 it has been predicted whether the motivation of a specific type increases or decreases in a certain development phase. These predictions are again confronted with the collected data (see Figure 195). In the following it is computed how many times these predictions are correct respectively wrong (see Figure 196). Probability statistics informs us how likely the number of correct predictions is produced by pure chance. Thereby it can be tested whether the overall hypothesis concerning the contingency of motivation on development phases is supported or not. This statistical procedure to test hypothesis 39a concerning motivation and development phases is similar to the one of Bales and Strodtbeck (see also Bales 1950; 1960).

Relationships between motivation and development phases					
			Development phases		
			p^1 ➔ p^2	p^2 ➔ p^3	p^3 ➔ p^4
			Initial phase	*Intermediate phase*	*Mature phase*
Motivation	Fun seeking	*Prediction*	Decreasing	Decreasing	Decreasing
		Reality	Increasing	Decreasing	Increasing
	Pro social	*Prediction*	Increasing	Same	Decreasing
		Reality	Increasing	Increasing	Decreasing
	Peer recognition	*Prediction*	Decreasing	Same	Increasing
		Reality	Decreasing	Increasing	Increasing
	Personal requirements	*Prediction*	Increasing	Increasing	Increasing
		Reality	Increasing	Decreasing	Increasing
	Reputation	*Prediction*	Increasing	Increasing	Increasing
		Reality	Increasing	Increasing	Increasing
	Human capital	*Prediction*	Increasing	Decreasing	Increasing
		Reality	Increasing	Decreasing	Increasing

Figure 195: Motivation and development phases: Prediction and reality

If one assumes that the probability that a motivation increases or decreases is 50% each, nine correct and wrong predictions can be expected. In fact, from 18 predictions 13 are correct that is significantly better than mere guessing.[713] Guessing whether motivation increases or decreases produces on average similar or better predictions in 2.5 cases out of 100 attempts. In consequence, the found pattern is significant and hypothesis 39a is supported.

[713] Chi-square test with 1 *df*, P = 0.025.

Wrongly and correctly predicted relationships								
		Development phases						
		p^1	➔	*p^2*	➔	*p^3*	➔	*p^4*
			Initial phase		*Intermediate phase*		*Mature phase*	
Motivation	Fun seeking		X		✓		X	
	Pro social		✓		X		✓	
	Peer recognition		✓		X		✓	
	Personal requirements		✓		X		✓	
	Human capital		✓		✓		✓	
	Reputation		✓		✓		✓	

X = wrong prediction ✓ = correct prediction

Figure 196: Motivation and development phases: correct and wrong predictions

Motivation and development phase: Functions

In chapter 8.2.6 several functions were specified to describe how the different types of motivation change contingent on the development phase. Four different generic functions associated with the made assumptions were set up:

i. Linear function with two parameters (fun seeking, personal requirement): $\underline{ax + b}$

ii. Logarithmic function with two parameters (reputation): $\underline{a\ln x + b}$

iii. Second order function with three parameters (pro-social motivation, peer recognition): $\underline{ax^2 + bx + c}$

iv. Third order function with four parameters (human capital): $\underline{ax^3 + bx^2 + cx + d}$

Motivation and development phase: Nonlinear regressions are run

In the following each of the four functions presented above are tested to find out whether the assumptions presented in chapter 8.2.6 fit the data. To test the prediction for each motivation nonlinear regressions with all the four generic functions were run. It was analyzed how good the dependent variable "motivation" can be predicted by the independent variable "phase development". This computation results for each motivation type and for each generic function in a residual sum of squares. The residual sum of squares is the amount of variability that cannot be accounted for in the regression model. The lower the residual sum of squares the better the function fits the data. Clearly, out of the four generic functions the one specified in chapter 8.2.6 should fit the data best. For example: For investment in reputation the nonlinear regression with a logarithmic function should produce a lower residual sum of squares than the ones with the remaining three functions.

Motivation and development phase: Nonlinear regression and information criterions

The problem with the above suggested procedure of hypothesis testing is that a model with more parameters and of a higher order cannot be worse than a model with less parameters and of a lower order. For example if a relationship between two variables is completely linear the function $ax + b$ fits the data perfectly. However, a second order function such as $ax^2 + bx + c$ fits the data similar good because if "a" is zero the two functions are similar. In practice, a second order function fits the data always better than a linear function. The fit of any model can be therefore improved by increasing the number of parameters (Burnham & Anderson 2002: 31). Indeed, for all types of motivation the residual sum of squares is the lowest if the regression is computed by a third order model.

Yet, overfitting a model should be avoided (Burnham & Anderson 2002: 29). Akaike (1974: 716) argues that "statistical identification is essentially concerned with the art of approximation". This can be described by an example of Wel (1975, reported in Burnham & Anderson 2002: 29-30). Wel posed the question how many parameters it takes to fit an idealized elephant that is drawn with 36 points. If the elephant is drawn with five or ten parameters there is no resemblance with the original elephant.[714] The drawing with 20 parameters is close enough to recognize it as a mammal standing on four legs. If the drawing has 30 parameters it is quite similar to the original elephant. Using additional parameters improves the drawing and if 36 parameters are used the fit is perfect. However, the improvement of using more than 30 parameters is marginal. How many parameters should be used in this example is basically based on the question at which point an elephant is accepted as an elephant.

The example of Wel shows the conflict between increasing model fit and increasing the number of parameters. Adding more parameters without improving the fit significantly contradicts the principle of parsimony in statistics (Burnham & Anderson 2002: 29; Rost 2004: 89-90). This is in line with Ockham's razor that advises as a heuristic to maxim simplicity and parsimony in argumentation. Therefore, a comparison of different models must include some sort of criterion that corrects for the number of parameters.

In statistics there are different methods to correct for the number of parameters (Burnham & Anderson 2002). The most known as well as the oldest criterion that corrects for the number of parameters is the **A**kaike **I**nformation **C**riterion (AIC, Akaike 1973). Other examples of information criterions are the Schwarz Information Criterion also known as the **B**ayesian **I**nformation **C**riterion (BIC, Schwarz 1978), and the **T**akeuchi **I**nformation **C**riterion (TIC, Takeuchi 1976). All these information criterions punish for including additional parameters and compare the complexity of a model with its goodness of fit. However, there are also differences between the mentioned information criterions. The BIC punishes an additional parameter more severely than the AIC. Rost (2004: 344) argues that the AIC should be used if there are few items and the BIC if there are many items in the model. The AIC and the BIC are based on different assumptions concerning the philosophy of science. While the BIC assumes that there is a true model the AIC does not (Burnham & Anderson 2004). The BIC is mostly used in sociology while the AIC is primarily used in econometrics.

A selection of a proper information criterion based on such subtle differences is beyond this study. The AIC is chosen based on practical reasons. It is easy to compute, effective in a wide variety of applications (Burnham & Anderson 2004: 270), and well known (Burnham & Anderson 2004: 263).

---

[714] Wel used least squares Fourier sine series (Burnham & Anderson 2002: 30).

AIC and AICc

According to Burnham and Anderson (2002: 66) the AIC does not perform adequately if there are too many parameters in relation to the size of the sample. To correct for small samples there is an adjustment called AICc. AICc should be used if the division of the sample size by the number of parameters is below 40. This applies for some models computed in this study. However, because AICc converges to AIC as the sample size gets larger the AICc can or should be used independent of the sample size (Burnham & Anderson 2004: 270).

The AICc measures cannot be directly compared and an individual AICc is by itself not interpretable. AICc measures are strongly dependent on sample size (Burnham & Anderson 2002: 70-72). Therefore, the relative AICc values, particularly the AICc differences, and not the absolute AICc values are important. To compute the AICc differences one has to subtract from each AICc value included in the comparison the one with the lowest value. Akaike (1981: 42) notes that “a model with a lower value of AIC is considered to be a better model”. The preferred model is the one with the lowest AICc value. As a rough rule of thumb a model with a difference smaller than two compared to the best model has substantial empirical support, models with differences between four and seven have considerably less support, and models with differences higher than ten are essentially not supported.

The formula of the AIC is (Burnham & Anderson 2004: 268):	$AIC = -2\log(L(\hat{\theta}\mid data)) + 2K$ respectively $AIC = n\log(\hat{\sigma}2) + 2K$ $K =$ *number of parameters* $L =$ *likelihood function*
The formula of the AICc is (Burnham & Anderson 2004: 66):	$AICc = AIC + \frac{2K(K+1)}{n-K-1}$ $n =$ *sample size*

Hypothesis 39b

In the following hypothesis 39b is tested. Figure 197 presents the AICc for all types of motivation and for all four generic functions:

- The best model for fun seeking is the logarithmic function. The linear function assumed to describe the distribution of fun seeking over the development phases best performs almost similar (AICc difference below 2). The corresponding assumption in chapter 8.2.6 has therefore substantial empirical support.
- The best model for pro-social motivation is the logarithmic function. The second order function assumed to describe the distribution of pro-social motivation over the development phases best performs middling compared to the best model (AICc difference below 2). The corresponding assumption in chapter 8.2.6 is therefore slightly supported.

- The best model for peer recognition is the second order function. This function was assumed to describe the distribution of peer recognition over the development phases best. The corresponding assumption in chapter 8.2.6 is therefore clearly supported.
- The best model for personal requirements is the logarithmic function. The linear function assumed to describe the distribution of personal requirements over the development phases best performs almost similar good compared to the best model (AICc difference almost zero). The corresponding assumption in chapter 8.2.6 has substantial empirical support.
- The best model for investment in reputation is the linear function. The logarithmic function assumed to describe the distribution of investment of reputation over the development phases best performs almost similar good compared to the best model (AICc difference almost zero). The corresponding assumption in chapter 8.2.6 has substantial empirical support.
- The best model for human capital is the linear function. The third order function assumed to describe the distribution of human capital over the development phases best performs not very good compared to the best model. The corresponding assumption in chapter 8.2.6 is not supported.

As a conclusion it can be noted that describing the distribution of motivation contingent on development phases by a function has only been completely successful for peer recognition. However, the assumptions related to fun seeking, pro-social motivation, investment in reputation, and personal requirements have substantial empirical support. The assumption related to human capital is not supported. Overall it can be concluded that hypothesis 39b is to some extent supported.

Motivation	Model	AIC	AICc	AICc diff
Fun seeking	$ax^3 + bx^2 + cx + d$	-100.44	-100.17	1.84
	$ax^2 + bx + c$	-100.39	-100.23	1.78
	aln(x) + b	-102.09	-102.01	0.00
	ax + b	-101.88	-101.80	0.21
Pro social	$ax^3 + bx^2 + cx + d$	-96.21	-95.93	3.55
	$ax^2 + bx + c$	-97.73	-97.56	1.92
	aln(x) + b	-99.56	-99.48	0.00
	ax + b	-99.34	-99.25	0.22
Peer recognition	$ax^3 + bx^2 + cx + d$	-24.25	-23.80	1.68
	$ax^2 + bx + c$	-25.75	-25.48	0.00
	aln(x) + b	-25.51	-25.38	0.10
	ax + b	-25.42	-25.29	0.19
Personal requirements	$ax^3 + bx^2 + cx + d$	-33.93	-33.66	4.08
	$ax^2 + bx + c$	-35.80	-35.64	2.10
	aln(x) + b	-37.82	-37.74	0.00
	ax + b	-37.78	-37.70	0.03
Reputation	$ax^3 + bx^2 + cx + d$	14.38	14.66	4.18
	$ax^2 + bx + c$	12.38	12.55	2.07
	aln(x) + b	10.49	10.57	0.09
	ax + b	10.40	10.48	0.00
Human capital	$ax^3 + bx^2 + cx + d$	-128.65	-128.38	2.37
	$ax^2 + bx + c$	-129.50	-129.34	1.41
	aln(x) + b	-130.69	-130.61	0.14
	ax + b	-130.83	-130.75	0.00

= Best model

Figure 197: Information criterion and hypotheses

11.4.7 Development phases and governance structures respectively mechanisms

Hypothesis 40

In hypothesis 40 it was suggested that the fraction of projects that are included in a major Linux distribution increases during development. In the initial phase the fraction of projects that are included in a major Linux distribution increases very slightly from 29.2% to 29.4% and remains therefore practically similar. In the intermediate phase this fraction increases significantly to 51%.[715] Finally, in the mature phase the fraction of projects included in a major Linux distribution increases significantly to 72%.[716] Hypothesis 40 is therefore partly supported.

Hypotheses 41 to 48

In hypothesis 41 it was suggested that during project development it gets increasingly difficult for an outsider to get code merged. In hypothesis 44 to 45 it was assumed that a project is increasingly missed by the entire community and the users if it were to disappear. It was also suggested in hypotheses 42, 43, and 46 that the number of developers, the involvement of commercial firms, and the number of different means to credit increases with project development. Hypotheses 47 and 48 assumed that during project development autonomy and the degree to which a project would be missed by the developers if it were to disappear decreases.

As suggested the difficulty to get code merged increases significantly with development going on (see Figure 198 and Figure 199). The involvement of companies and the number of programmers increases significantly with development as well. The number of modes to credit increases significantly with development going on although the relationship is not very big. Projects are increasingly and significantly missed by users with development going on.

Against the assumption projects are not decreasingly missed by developers. In fact, the opposite is true. Projects are also not increasingly missed by the entire community. Similarly, autonomy is not related to development phases. The degree of autonomy remains almost constant over the different development phases (see Figure 199).

In consequence, hypotheses 41, 42, 43, 45, and 46 are supported. Hypotheses 44, 47, and 48 are not supported.

[715] Chi-square test with 1 *df*, $P = 0.025$.
[716] Chi-square test with 1 *df*, $P = 0.025$.

Structures and mechanisms and development phases									
		Easy get code into project	Companies recently	Number of program-mers	Missed by community	Missed by developers	Missed by users	Number of modes to credit	Autonomy
Development phases	Correlation[a]	-0.12	0.11	0.30	0.02	0.09	0.23	0.09	-0.02
	P	0.0367	0.0497	0.0000	0.3754	0.0867	0.0006	0.0914	0.3889
	N	158	156	140	148	153	153	160	151

Figure 198: Structures and mechanisms and development phases

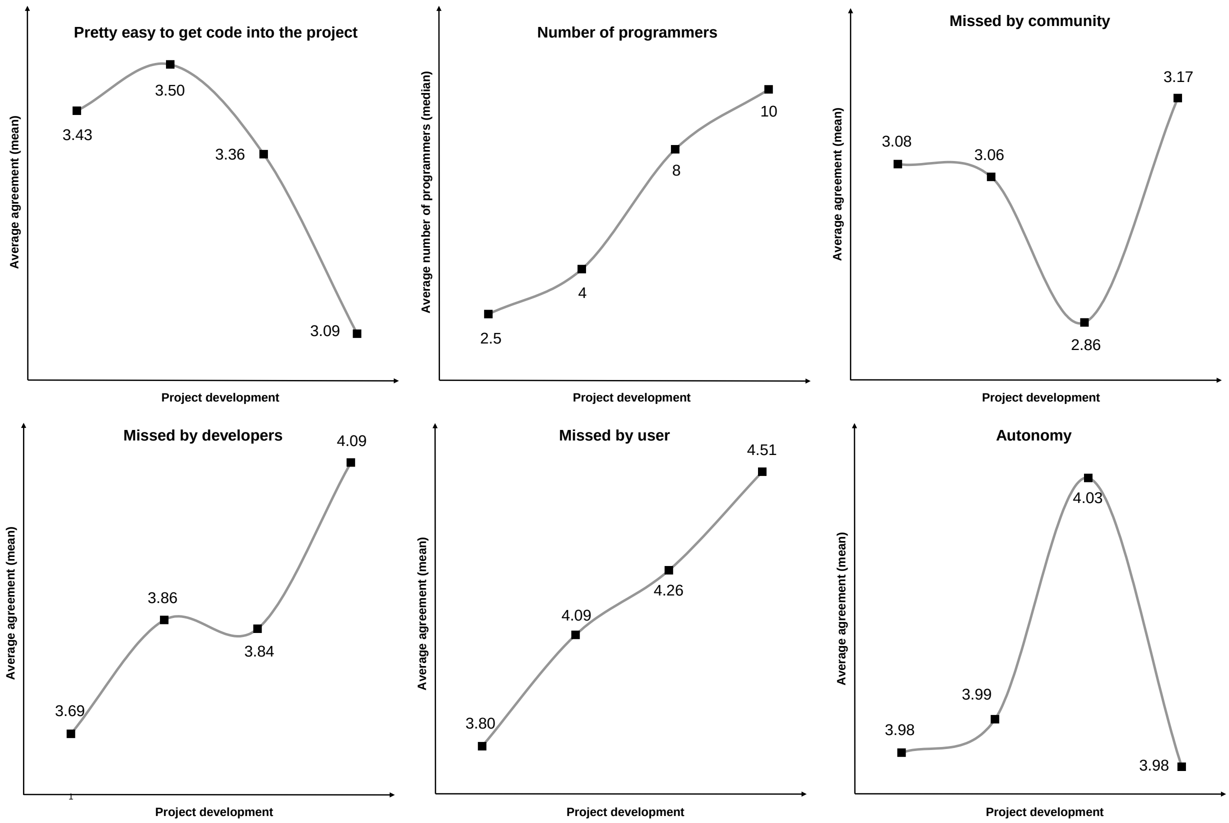


Figure 199: Structures and mechanisms: Means in different phases[717]

Hypotheses 49 to 50

In hypothesis 49 it was suggested that in the initial development phase of a project the fraction of projects that have already chosen a license increases. Indeed, in the initial phase this fraction increases from 81% to 85%. It increases to 92% in the intermediate phase and to 96% in the mature phase. Although the pattern is very clear the differences are not significant and hypothesis 49 is therefore not supported.

Hypothesis 50 assumed that in the initial development phase of a project the fraction of projects that have at least one mechanism to credit increases. However, there is no significant difference and hypothesis 50 is not supported.

---

[717] The “huge” increase in autonomy in the intermediate phase is artificially produced. In fact, autonomy remains constant.

Hypotheses 51 to 52

In hypothesis 51 it was suggested that received feedback increases in the initial phase, decreases in the intermediate phase, and increases again in the mature phase. Indeed, feedback increases in the initial phase from 3.11 to 3.41, decreases in the intermediate phase to 3.28, and increases again to 3.36. However, the change in feedback is not significant. Hypothesis 51 is therefore not supported.

It was also assumed in hypothesis 52 that the degree decisions are explained changes during development similar to feedback giving. Again, the distribution follows the assumption. In the initial phase the degree decisions are explained increases from 3.37 to 3.50, decreases in the intermediate phase to 3.28, and increases to 3.36 in the mature phase. These changes are not significant and hypothesis 52 is therefore not supported.

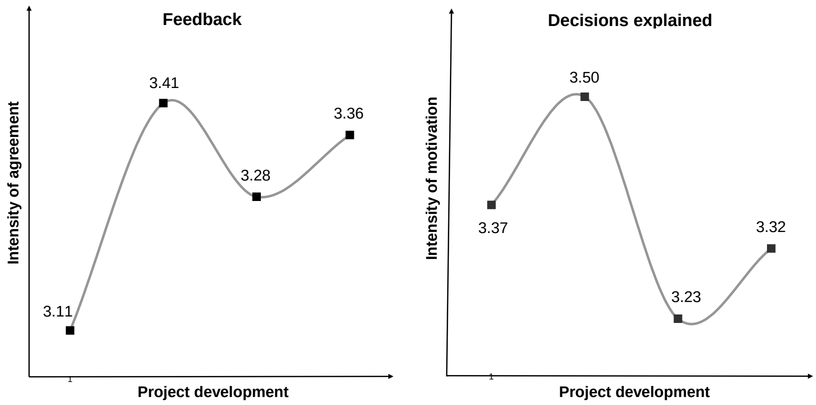
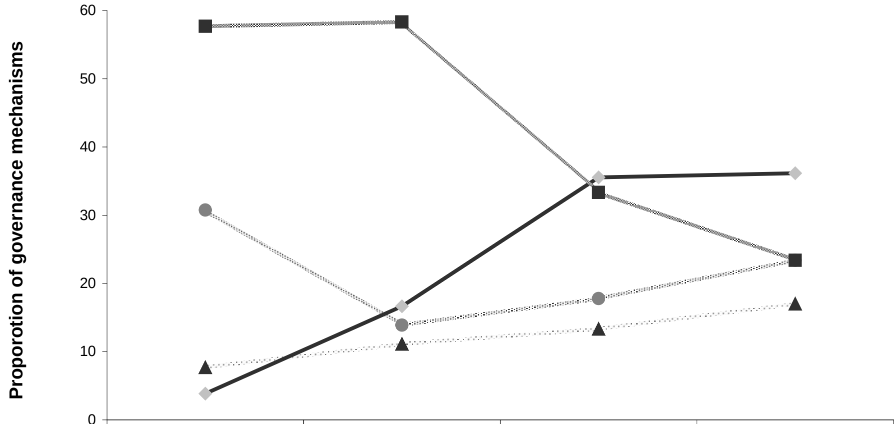


Figure 200: Structures and mechanisms (2): Means in different phases

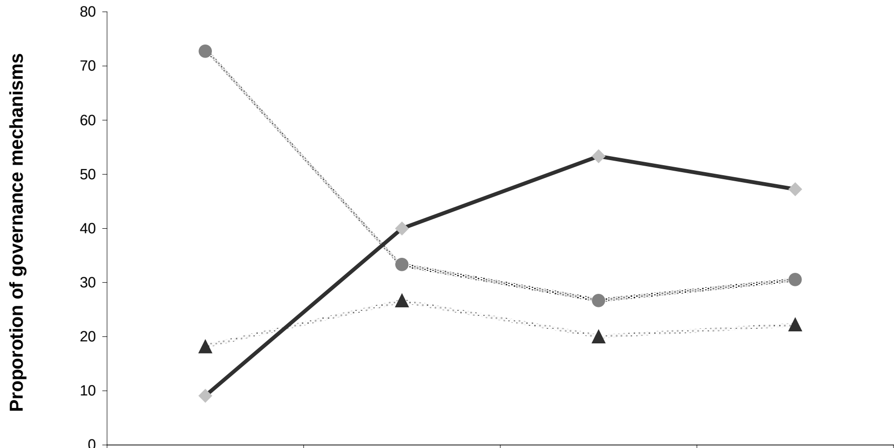
Decision making mechanisms

Besides hypothesis testing a notable observation concerning decision making mechanisms in different development phases should be mentioned. Data shows that over all phases 13% of the projects assign special positions by elections, 21% by appointments, 26% by informal consent, and that 40% of projects were too small to have related mechanisms. This pattern changes in different development phases. The proportion of projects that are too small for having special decision making mechanisms decreases with ongoing development (see Figure 201). If only projects are analyzed that are not too small it can be found that during development informal consent increases in importance while appointment decreases in importance (see Figure 202). Interestingly, the importance of elections remains practically constant during different development phases. It can be assumed that i) choosing elections as a decision making mechanism is a principal choice or ii) that elections are always efficient irrespective. It would be interesting to analyze the above mentioned observation. It may be that there are differences between the free and the open Source community.



	Planning / Proof of concept	Basic features complete	Ready for end-users	Feature complete / mature
Elected	7.69	11.11	13.33	17.02
Appointed	30.77	13.89	17.78	23.40
Informal consent	3.85	16.67	35.56	36.17
Too small	57.69	58.33	33.33	23.40
	N= 26	*N= 36*	*N= 45*	*N= 47*

Figure 201: Decision making mechanisms and development phases



	Planning / Proof of concept	Basic features complete	Ready for end-users	Feature complete / mature
Elected	18.18	26.67	20.00	22.22
Appointed	72.73	33.33	26.67	30.56
Informal consent	9.09	40.00	53.33	47.22
	N= 11	*N= 15*	*N= 30*	*N= 36*

Figure 202: S Decision making mechanisms and development phases (without “too small”)

11.5 Summing up and short discussion

In the following, the results are summed up and shortly discussed (see also Appendix IV).

11.5.1 Descriptive statistics

In short, the following results concerning **<u>FOSS projects</u>** were found:

- ***Company contributions:*** Companies hardly make important contributions at the beginning of a FOSS project. With development going on the involvement of companies gets increasingly important. This supports clearly the assumption that the motivation of actors to contribute is contingent on the development phase. It can be assumed that companies do not contribute at the start of FOSS projects because i) success is unpredictable and because ii) companies rate success as being very important.

- ***Project positions:*** In 24% of the projects special positions are assigned by elections, in 34% by appointment, and in 43% there is an informal consent. Interestingly, the importance of elections grows with project size but not with ongoing development.
- ***Number of programmers:*** On the average there are 5 developers contributing to a FOSS project. Only 30% of the projects have more than 10 and only 17% of the projects have more than 25 developers. The number of developers is highly dependent on the development phase. Projects at the beginning report on the average 2.5 and mature projects 10 developers.
- ***Development phases:*** There are roughly one-fourth of projects in each of the four different development stages adopted here.
- ***Credits:*** The most frequently used instruments for giving contributors credits are websites and credit files. There are virtually no projects that omit giving credits in some way. This highlights clearly the importance to credit contributors and acknowledge their work. However, this analysis suggests very strongly that giving credits has nothing to do with producing valuable signals for outsiders but with self-esteem, hacker ethic, and pride.
- ***Licenses:*** About 89% of the projects are licensed under copyleft terms and roughly 81% of the projects are subject to *strong* copyleft licenses. Thereby the importance of strong copyleft licenses respectively the GNU GPL, the most import representative of a strong copyleft license, is clearly corroborated. Developers are almost unanimously happy with copyleft and strong copyleft licenses.
- ***Inclusion in major Linux distribution:*** One half of the projects are included in a major Linux distribution while the other half is not.

The **FOSS developers** included in the survey had the following characteristics:

- ***Community affiliation:*** One-third of the subjects included in the survey belong to the free software community, one third to the open source community, and one third does not care to which community they belong.
- ***Satisfaction with FOSS involvement:*** Looking back most subjects consider their participation in a FOSS project as a good decision. In general, it can be concluded that FOSS developers are quite happy with their FOSS participation.
- ***Project roles:*** A surprisingly high fraction of one-fifth of the contributors to FOSS projects are not engaged in some sort of coding. This urges to analyze such individuals specifically. It also supports the assumption the FOSS phenomenon may be applied beyond software development. If individuals engage in translating and the like in FOSS projects it seems to be possible that individuals also engage in translating in other domains.
- ***Project experience:*** On the average the project experience of the subjects is roughly 3 years.
- ***Working hours for project:*** The subjects in the study work on average five hours (median) respectively ten hours (mean) per week for their project. This is clearly a huge effort. At the same time this finding is comparable to volunteer work in general. In Germany 47% of male volunteers (female volunteers 32%) donated more than five hours a week (BMfSFJ 2005: 397).

- ***Paid project members:*** About one-tenth of project members are to some extent paid. This clearly challenges the frequent claims in science and the media that FOSS applications are *solely* produced by unpaid developers. Paid FOSS members work on average considerably more hours for their project.
- ***Business interests:*** The contribution of one-third of all subjects is connected to some business interests. From these subjects one-third is paid for project participation by an employer and two-thirds have an own business related to the project. The total amount of contributed hours per week of employed and self-employed developers is almost similar. Business interests are related to the implementation of special functionalities and bug fixing. To maintain a good reputation within the FOSS community is considerably less important. Adjusting software to specific needs seems to be more important for commercial firms than for individual developers (see personal requirement in chapter 11.4.1).
- ***Programming experience:*** The subjects have a very high programming experience of about 14 to 15 years on average. Therefore, it may be stated that FOSS development is dominated by professionals with a huge experience and not by freshmen who learn to code.
- ***Working hours:*** Apart from FOSS project participation developers (students included) work on average 40 hours a week in a paid job. Roughly 29% (46% if students are excluded) of the subjects work 41 or more hours per week in a paid job. This is comparable to the 35% of all workers in the US in 2004 who worked more than 41 hours (Lee, McCann, & Messenger 2007: 212). However, taken into account that software engineers (BLS 2006b) and computer programmers (BLS 2006a) usually work long hours it may be assumed that FOSS developers work less in paid jobs than their peers not involved in FOSS development.
- ***Education:*** More than four-fifth of the subjects have completed the entry requirement for university and three-fifth of the subjects have some sort of university degree. The educational level of the subjects is therefore very high. At the same time this education level is comparable to computer programmers in general (BLS 2006a). However, the fraction of developers with a master degree is considerably higher for FOSS developers than for all computer programmers in the US. However, this result may be (but not necessarily) an artifact caused by sampling.
- ***Students:*** More than one-third of the subjects are students. Approximately two-thirds of the students study something related to IT.
- ***Occupation:*** The occupation of almost one-fifth of the subjects has no IT relationship. Subjects with an occupation related to software are mostly concerned with development, administration, or support. The few subjects whose occupation is related to hardware are primarily occupied with sales.
- ***English proficiency:*** The subjects consider their English proficiency as good. Observation of the subjects filling out the survey suggests partly otherwise.
- ***Gross income:*** The average monthly gross income of the subjects is about 2000 US$.

11.5.2 Items and scales

In general, the *items* included in the survey are thought to be sound. However, the used *scales* performed only middling. Especially the following scales are not adequate:

- ***Peer recognition:*** The motivational scale related to peer recognition performed poor. A major cause for this may be the misadventure already reported in chapter 11.1. In the survey version printed for the FOSDEM conference one word of an item related to peer recognition cleared out. Thereby the content of the question was not interpretable and related answers were coded as missing. This resulted in 42% missing values for the item in question. Another problem was the wording of another item measuring peer recognition. The other motivation scales performed better but clearly not perfect.
- ***Basic need satisfaction:*** The scales related to basic need satisfaction did scarcely produce reliable scores. Major problems were the negatively worded items. For reasons not completely understood all negatively worded items seemed to have led to confusion (see chapter 11.3.1.2).

The generally low reliability of the scales seems to have two antecedents. *Firstly*, the number of items used to form the scales was very low. Using only few items for measuring the scales was a deliberate choice. To some extent a low reliability must therefore be accepted. Adding more items solely for increasing reliability has certainly major disadvantages. *Secondly*, the English proficiency of subjects seems to have been overestimated. Reliability increases for several motivation scales to levels above 0.7 if only subjects with a very good English proficiency are included. In consequence, it would probably have been beneficial to translate the survey in other languages such as German for the Chemnitz cluster and French for the Brussels cluster.

Because of the discussed problems the results related to scales have to be judged with suspicion.

11.5.3 Basic idea

A pattern of results is more meaningful than a single statistically significant finding (Sauley & Bedeian 1989: 340). The pattern found in this study seems to be quite clear: development phases are related to developer motivation. The basic idea that the motivation of developers is contingent on the development phase seems to be very sound.

11.5.4 Motivation

It was hypnotized that fun seeking and investment in human capital are the two most important types of motivation and that investment in reputation is the least important motivation. These hypotheses proved to be correct. The two most important types of motivation are fun seeking and investment in human capital. Less important are pro-social motivation and personal requirement. The two least important types of motivation are peer recognition and investment in reputation. Interestingly, this corresponds to the results of the BCG study. Again, it should be noted that no probabilistic sample has been drawn. Therefore, all generalizations are error prone. The results presented above should therefore be only accepted with a grain of salt.

The results presented here and in other studies suggest strongly that the importance of investment in reputation is marginal for FOSS development. The prominence of this type of motivation in academic discussions about FOSS is not found in reality.

Fun seeking and investment in human capital are basically related to all other types of motivation respectively to all types of motivation except investment in reputation. Reputation investment is only related to fun seeking.

It was suggested that fun seeking, pro-social motivation, and peer recognition group while personal requirement, personal requirement, investment in reputation, and investment in human capital group as well. This proved to be correct. It can therefore be concluded that using self-determination theory to research the motivation of FOSS developers is appropriate.

11.5.5 Motivation and governance structures respectively mechanisms

Decision making procedures

The analysis of the relationship between the assignment of project positions and developer motivation did not produce the expected results. It was found that:

- ***Fun seeking***, ***peer recognition***, and ***personal requirement*** are the highest if special project positions are ***appointed hierarchically***.
- ***Pro-social motivation*** is the highest in projects that either assign special positions by ***informal consent*** or that ***appoint them hierarchically***.
- The motivation related to ***human capital*** is the highest in projects that conduct ***elections*** or in projects that assign positions by ***informal consent***.
- The motivation related to ***investment in reputation*** is the highest in projects that conduct ***elections***. The relationship is, however, non-significant.

That investment in reputation is related to elections was hypothesized. Therefore this is result can be theoretically interpreted. For all other results the explanation is theoretically not easy.

Fun seeking and governance structures respectively mechanisms

Astonishingly, fun seeking is not related to autonomy or relatedness. That autonomy is not related to fun seeking may be theoretically explained by the possibility to stop contributing to a FOSS project immediately if developers feel restricted. Developers motivated by fun seeking are either happy with the degree of autonomy or quit very fast. With this explanation the importance of autonomy is not neglected. However, the consequences of low autonomy are limited to the refusal to contribute due to the easily exerted exit option.

Competence, feedback, and the degree a project would be missed by developers if it were to disappear are, as suggested, positively related to fun seeking.

Pro-social motivation and governance structures respectively mechanisms

The analysis showed how important the license of a FOSS project is for pro-socially motivated developers. Pro-social motivation is higher if subjects are happy with the license of a project. More precisely, pro-socially motivated individuals are motivated by strong copyleft licenses. While this result is probably not surprising it is nevertheless theoretically and practically very important.

Peer recognition and governance structures respectively mechanisms

Peer recognition was found to be positively correlated to relatedness as suggested but not to the number of programmers in a project and the degree a project would be missed by developers and the community if it were to disappear.

Personal requirement and governance structures respectively mechanisms

Personal requirement is positively related to the degree a project would be missed by the users if it were to disappear. This result is according to the suggestions presented in chapter 8.2.3. Rather surprising is the observation that the inclusion of a FOSS application in a major Linux distribution does not influence motivation related to personal requirements. It was argued that such an inclusion is very convenient for a user and saves a lot of time. Especially for a user interested in the efficient use of applications the inclusion in a distribution should be of major concern. No satisfying explanation for this result was found.

Investment in reputation and governance structures respectively mechanisms

The hypothesized relationships between investment in reputation and governance structures respectively mechanisms are corroborated by the data. Investment in reputation was found to be positively related to the number of developers in a FOSS projects and to the degree a project would be missed by its users and the community if it were to disappear. Two additional findings seem to be of special interest. *Firstly*, investment in reputation is positively related to the contributions of companies. This contradicts the usual argument about investment in reputation that implicitly assumes that it is in FOSS projects always possible to produce valuable signals. The data may be interpreted insofar as signals can only be produced if there is an audience respectively if commercial firms are engaged in a FOSS project. *Secondly*, the number of means to credit is positively related to investment in reputation. Projects that intend to attract individuals motivated by investment in reputation should have different means to credit. Having some sort of mechanisms to credit contributions is standard. There are virtually no projects without at least one mechanism to credit.

Human capital and governance structures respectively mechanisms

Human capital is positively related to feedback, competence, and the degree a project would be missed by the developers if it were to disappear.

11.5.6 Paid developers and the involvement of firms

Paid developers contribute more likely to projects to which commercial companies contribute as well. This result is not surprising but nevertheless interesting.

The question of undermining intrinsic motivation respectively fun seeking by an external intervention such as paying for contributions was skin-deep researched. The data does not support the assumption that fun seeking is undermined by the payment for FOSS contributions. An explanation may be that paid developers still decide on their own what and how they contribute. The relationship between sponsor and contractor may be different in FOSS development than the one found in other areas. Anecdotal evidence suggests that this may be the case indeed. However, corresponding further research is needed.

It was suggested that individuals motivated by personal requirement, pro-social motivation, and investment in reputation do not contribute in an early development phase because project success is hardly foreseeable. From the 120’000 projects hosted at SourceForge most will not succeed or remain non-functional. To wait which of the projects succeed and which vanish is therefore a reasonable strategy. This argumentation does also hold true für the involvement of commercial firms. Indeed, it was observed that the involvement of commercial firms increases during development.

11.5.7 Interaction of motivation and governances structures respectively mechanisms

The interaction hypothesis discussed in chapter 8.2.3 is not supported by the data. The interaction variables increase the explained variance by 7% for fun seeking, 8% for pro-social motivation, 10% for peer recognition, 1% for personal requirement, 1% for human capital, and 1% for reputation investment. The increase is for all motivation types not significant. An easy explanation why the hypothesis is not supported could not be found. Perhaps the related effect is too small, the methodology is not suitable, or the theoretical thoughts are wrong.

11.5.8 Motivation and type of engagement

The relationship between motivation and the type of engagement is highly relevant for practical purposes. If such relationships are known governance structures and mechanisms could be adjusted to the type of engagement.

Basically, the results showed that there is a relationship between motivation and the type of engagement although the specific hypotheses were not always supported. It was found that:

- that ***fun seeking*** is positively related to ***adding/developing features*** and to ***review source code,***
- that ***pro-social motivation*** is positively related ***marketing/promoting*** but not to mundane tasks such as translating, creating graphics, writing FAQ’s, and donating,
- that ***peer recognition*** is not related to any type of engagement,
- that ***personal requirement*** is positively related to ***adding/developing features*** but very surprisingly not to fixing or reporting bugs,
- that ***investment in reputation*** is positively related to ***promoting/marketing,*** and
- that ***human capital*** is positively related to ***review source code*** and to ***adding/developing features*** (not significant).

Four observations are remarkable:

Firstly, pro-socially motivated developers do not engage in mundane tasks. It was suggested that pro-social developers engage in such tasks because i) nobody else does and ii) because they are necessary for project success. The question emerges: if not pro-socially motivated developers who else engage in mundane tasks? Some arguable suggestions maintain that translating and writing FAQ’s are positively related to investment in human capital, that creating graphics is positively related to personal requirements, and that donating is positively related to peer recognition.

Secondly, the motivation related to human capital is significantly higher for subjects who review source code. This supports the results presented in chapter 3.2.2.3 about how one learns in FOSS projects: reading other people's source code.

Thirdly, personal requirement seems not to be related to bug fixing or bug reporting. This is very surprising because fixing bugs or getting bugs fixed seem to be very important for using an application efficiently. Furthermore, the effort to report an annoying bug seems to be low. Developing a new feature is clearly more costly in terms of time. No reasonable theoretical explanation could be found for this observation.

Fourthly, peer recognition is not related to any type of engagement. This could be explained by the argumentation that being embedded in social relations does not depend on the executed task.

11.5.9 Motivation and community affiliation

The results of this study support the assumption that being part of the free software community is related to pro-social motivation and to projects with strong copyleft licenses. This highlights the importance of community affiliation for FOSS research.

11.5.10 Motivation and development phases

The results concerning the influence of development phases on motivation are so-so:

- ***Very good***: The theoretical assumptions related to ***peer recognition*** in different development phases are clearly supported by the data. No major theoretical adjustments seem to be indicated.
- ***Fair:*** The empirical results related to ***personal requirement*** are only middling. Nevertheless, it seems to be that the theoretical assumptions are sound. It is suggested that the middling empirical results are due to methodological and not to theoretical problems. If the item "forking" is excluded from the corresponding motivation scale the relationship is almost completely linear. Feedback received primarily in Chemnitz suggests that some subjects did not know what forking is. Therefore, no major theoretical adjustments are indicated.
- ***Fair***: The empirical results related to ***investment in reputation*** and ***pro-social motivation*** are fair. In both cases adjustments to the theoretical assumptions seem to be straightforward. Investment in reputation does not seem to increase in the course of development logarithmically. Rather this type of motivation seems to increase linearly with development going on. Pro-social motivation seems to increase logarithmically during development.
- ***Fair***: The empirical results related to the assumptions concerning ***human capital*** are very good. However, the theoretical assumptions are not completely convincing and seem to need some refinement or further analysis.
- ***Poor***: The empirical results related to ***fun seeking*** are poor. Based on this observation the theoretical assumption about fun seeking in different development phases must be totally refuted and revised.

A result that needs some analysis is the distribution of the fun seeking scale contingent on FOSS development (see above). One line of reasoning may explain this observation by autonomy. Because autonomy does not decrease linearly as suggested but remains basically constant over the different development phases (see Figure 199) it can be assumed that fun seeking remains constant as well. An explanation why autonomy does not change during development sounds straightforward: Developers motivated by fun seeking choose the project they contribute to themselves. Due to this self-selection mechanism fun seekers do never feel that their autonomy is restricted by backward compatibility and the like. Such restrictions are then not considered to be controlling but to be an interesting challenge. Otherwise they quit contributing to the FOSS project in question and contribute to another project that does not restrict autonomy. However, because the psychometrics of the autonomy scale was poor methodological problems may also have caused the results. Therefore, researching fun seeking again with an improved autonomy scale would be necessary.

One note has to be made concerning the distribution of different types of motivation contingent on the development phase. That a motivation type is in a certain phase more important than in another phase does not tell anything about the importance of one particular motivation in contrast to other types of motivation. Pro-social motivation decreases in importance in the mature phase while investment in reputation increases in importance. However, independent of this observation pro-social motivation is by far more important in the mature phase than investment in reputation.

11.5.11 Development phases and governance structures respectively mechanisms

Project development influences also different governance structures and mechanism of FOSS projects. With development going on:

- the fraction of projects included in a major Linux distribution increases,
- the fraction of projects that have made a decision about licenses increases,
- the difficulty to get code merged increases,
- the involvement of companies increases,
- the number of involved programmers increases,
- the number of means to credit contributors increases,
- projects are increasingly missed by its users if they were to disappear,
- projects are increasingly missed by its developers if they were to disappear, and
- the importance of elections and particularly informal consent for the assignment of special positions in FOSS projects increases while the importance of appointment decreases.

The major characteristic that does not change during development is autonomy. The degree of autonomy remains surprisingly constant over the different development phases. Whether this is due to the reported methodological problems (see chapter 11.5.2) or due to a theoretical misspecification (see chapter 11.5.10) is a question of major interest that can unfortunately not be solved at this point.

12 Final thoughts: Lessons and outlook

In the introduction the resource based view and how firms can achieve a sustainable competitive advantage was discussed in order to explain why it is relevant to solve the problems related to collective action. It was argued that a sustainable competitive advantage rests on an optimal mix of resources that are valuable, rare, reasonably durable, and difficult to imitate or substitute. It was further stated that organizational knowledge meets these requirements because it is based on unique interactions between individuals, formal and informal interpersonal relationships, and a system of norms and beliefs. This acknowledges the fact that human actors do not live in a Crusoe economy.

However, successful interpersonal interactions, interpersonal relationships, and the establishment of norms and beliefs are not trivial to organize. Some of the problems are related to social dilemmas. They arise if i) individuals receive a higher payoff for a socially defecting choice than for a socially cooperative choice no matter what other individuals do and if ii) all individuals are better off if cooperation prevails over defection. Two factors influencing the solution of social dilemmas were given special attention in this dissertation:

i. ***Motivation***: Individuals with competitive or individualistic motives display in general higher non-cooperative behavior than the ones with altruistic or cooperative motives. In consequence, motivation matters if social dilemmas must be solved.

   In this dissertation motivation has been connected to governance structures and mechanisms of FOSS projects because they provide incentives and restrictions. Based on the interplay between motivation and governance structures respectively mechanisms FOSS developers decide whether they contribute to a project or not.

ii. ***Decision sequence***: Sequential or incremental decision making changes *firstly* payoffs. Payoffs may alter contingent on the decisions of others. This may be either based on a psychological desire to jump on a bandwagon or on positive network effects. *Secondly*, sequential decision making improves the information for the successor. Coordination dilemmas such as the one in the stag hunt game vanish if the game is played sequentially. *Thirdly*, it allows antecessors to make a credible commitment. *Fourthly*, sequential decision making allows conditional cooperation.

   In the course of this dissertation decision sequence is an important component because it introduces the notion that the decision situation changes contingent on the development phase. Not all development phases provide the same incentives or decision options. In consequence, it can be stated that dynamical forces matter.

Based on the interplay between motivation and governance structures respectively mechanisms collective action in FOSS projects has been analyzed. It was acknowledged that the decision sequence respectively dynamical forces are important for explaining collective action. In the following, the related research questions are discussed.

12.1 Research question: Discussion

In chapter 1.3 the following basic research questions related to the dynamics of collective action in general have been presented:

i. *Is the problem of collective action really shaped by dynamical forces?*

ii. *How are different types of motivation related to the dynamics of collective action?*

iii. *How are governance structures and mechanisms related to the dynamics of collective action?*

These questions were applied to FOSS development:

iv. *Is the FOSS innovation model shaped by different development phases?*

v. *Is there a relationship between motivation and ongoing FOSS development?*

vi. *How are governance structures and mechanisms related to FOSS development phases?*

In the following, the questions i. to vi. are answered bottom up. Questions vi. to iv. are discussed first. The corresponding insights are used to answer questions iii. to i as well.

Certainly, deducing from a special case, namely FOSS development, to a general phenomenon is not without pitfalls.[718] If it is found that dynamical forces play an important role in FOSS development it is not granted that this is also true for collective action in general. Nevertheless, it shows that at least under some conditions dynamical forces are relevant.

Again, it should be noted that a cross-sectional research as the one adopted here is not able to distinguish between age and cohort effects (see chapter 9.1).

12.1.1 How are governance structures and mechanisms related to FOSS development phases?

In chapter 4.3 a large but certainly not conclusive list of different governance structures and mechanisms used in FOSS development was presented. Because of limited space in the survey it was only possible to research a few FOSS structures and mechanisms.

There are several interesting results concerning the relationship of governance structures respectively mechanisms to FOSS development phases. Three of them are exemplarily discussed:

- As hypothesized the *involvement of companies* increases with ongoing development. The same holds true for the *number of developers* in a project. It can therefore be assumed that there is some sort of

---

[718] An interesting illustration that one may find an innumerable number of examples although there is no universal pattern is the prime number theorem that gives a rough description about the distribution of primes. For a very long time it was thought that the heuristic estimation of Gauss *overestimates* the real number of primes. It was found empirically that the estimation of Gauss is true for all numbers up to a million, a billion (the following numbers refer to the short scale), a trillion (10^{12}), a quadrillion (10^{15}), and a quintillion (10^{18}). However, based on the reasoning of Littlewood (Littlewood 1914) Stanley Skewes (1933: 288) argued that the upper bound for the smallest natural number for which the number of primes is *underestimated* is approximately $e^{e^{e^{79}}}$ respectively $10^{10.000.000.000.000.000.000.000.000.000.000.000}$ (please note that this upper bound has now been reduced drastically). If one compares Skewes' number with the number of protons and electrons in the universe which is thought to be each approximately 10^{79} (Eddington 1941: 170) one can imagine that by induction one would have never ever found that the estimation of Gauss may underestimate the real number of primes.

herding. There are at least two distinct but complementary explanations for this observation: *Firstly*, individuals like to contribute to a successful project that has already gained some fame within the community. Related to this psychological desire it may be that individuals like to be part of a bigger community even if communication is exerted electronically. *Secondly*, many actors refrain from contributing until success is manifest or at least predictable. They do not want to waste scare resources. In consequence, project success matters which in turn leads to the question who starts to contribute to a new FOSS project.

- Not surprisingly, the proportion of projects that are too small for having special *decision making mechanisms* decreases with ongoing development. For projects that are not too small *informal consent* as a mechanism increases in importance with ongoing development while *appointment* decreases. The importance of *elections* remains practically stable during ongoing development. There are at least two explanations for this last observation. *Firstly*, choosing elections as a decision making mechanism is based on a principal choice rooted in a moral concept irrelevant whether this decision making mechanism is efficient or not. *Secondly*, the efficiency of elections as a decision making mechanism does not considerably change in different development phases. In consequence, election is chosen equally in all development phases because it is equally efficient. On instinct, the former explanation seems to be clearly more plausible. If so, it seems to be important to adopt decision making mechanisms based on the motivation of the developers contributing to a project.

- An important component of the argumentation in this dissertation was that *autonomy* decreases during development. Contrary to this assumption it was found that autonomy remains more or less constant in all development phases. This result is of special importance because it is thought that autonomy is a necessary condition for fun seeking which in turn is a major motivation to contribute to FOSS development. Based on the empirical results it could be hypothesized that autonomy is always important in FOSS development. That the autonomy of developers remains constant is then connected to the self-selection mechanism found in FOSS development. If restrictions such as backward compatibility emerge in the course of development and if these restrictions are perceived as controlling developers stop contributing (see also the Mambo-Joomla! example discussed in chapter 2.7). However, if developers do not stop contributing restrictions are probably perceived as challenging and not as controlling. If this line of reasoning is true this empirical result has an interesting implication for commercial projects. Stop contributing as an exit option is in commercial projects not easy because of cancellation periods and mainly because of financial necessities. In consequence, commercial projects should not reproduce FOSS governance structures and mechanisms one-to-one. Commercial projects should rather give more attention to autonomy because the importance of the exit option, which serves as a valve, is lower in contrast to FOSS projects. If commercial firms do not give more attention to autonomy fun seekers may get demotivated. Alternatively, commercial projects that intend taking advantage of the FOSS production model may deliberately rely on other motivation types than fun seeking.

  However, the above presented argumentation is based on a doubtful empirical result. The psychometrics of the autonomy scale was very poor and not completely reliable. Therefore more research is needed to find out whether the observed result can be explained by theoretical or methodological reasons (or both).

Based on the above presented and discussed results it can be inferred that some governance structures and mechanisms are related to FOSS development phases while others are not.

12.1.2 Is there a relationship between motivation and ongoing FOSS development?

Motivation is of special interest for the analysis of FOSS because developers are usually not paid. It was argued that in community projects fun seeking, pro-social motivation, peer recognition, personal requirement, reputation investment, and investment in human capital are important types of motivations.

In a nutshell, it can be noted that the type of motivation is clearly related to the dynamics of collective action in FOSS development. [719] The relative importance of different motivation types changes during FOSS development. For example, based on the empirical results it seems to be obvious that investors in reputation contribute increasingly with FOSS development going on. The rationale is simple: Reputation investors strive to produce a valuable signal. The value of a signal is related i) to its visibility and ii) to its credibility. It is thought that visibility and credibility increase during FOSS development. Therefore, the motivation for reputation investors to contribute to a FOSS project is positively related to the development phase of a project.

12.1.3 Is the FOSS innovation model shaped by different development phases?

Decision sequence is of special importance in FOSS development because there are *firstly* huge network externalities and high switching costs. *Secondly*, the production functions of many FOSS applications are thought to be S-shaped. This leads, contingent on the development phase, to big differences in the value of distinct applications. Network externalities, switching costs, and the S-shaped production function are related to a change in payoffs provided for contribution in different development phases. *Thirdly*, there is a selection problem for prospective contributors. On the one hand the amount of FOSS projects available to contribute to is enormous. On the other hand most of the FOSS projects will not meet with success as long as user adoption or the like is taken as an indicator for success. For developers for which success is important this leads to a selection problem. Waiting with contributions until project success is foreseeable is a strategy to scope with this problem. This strategy is connected to the amount of information available in different development phases. *Fourthly*, decision sequence is important in FOSS development because project founders can make a very credible and important commitment: To choose the GNU GPL ensures that software remains free what is very important for pro-socially motivated developers. To do so is credible because it cannot be reverted.

Because of the reasoning presented above it can be concluded that different development phases are very crucial for FOSS development. In the course of this dissertation it was shown theoretically and with different examples that motivation and decision sequence respectively development phases are related to each other. This does also hold true for FOSS development. Different development phases of FOSS projects provide different incentives and restrictions. Shortly speaking the following line of reasoning was proposed concerning the relationship between motivation and development phases:

[719] This conclusion does not imply that the importance of the motivation types have always changed according to the hypotheses. While the corresponding assumptions for peer recognition were very good the ones for fun seeking were poor.

Fun seeking

- *Assumptions concerning motivation*: Fun seekers contribute unconditionally because they enjoy producing source code. They do not care about the success of a project. However, if autonomy decreases developers stop contributing.
- *Assumptions concerning decision sequence*: With development going on the autonomy in a project decreases constantly.
- *Assumed consequence*: Fun seekers contribute decreasingly with development going on.
- *Appraisal*: The empirical results are poor. The theoretical assumption about fun seeking must be totally refuted and revised.

Pro-social motivation

- *Assumptions concerning motivation*: Pro-socially motivated developers are interested in the advancement of FOSS that is primarily achieved by successful projects. They contribute therefore only to projects with some likelihood of being successful. At the same time if other individuals assure project development they quit contributing and concentrate on projects desperately in need for contributions. Pro-socially motivated developers will not contribute to a project that has not yet made a commitment that it remains free such as putting the source code under the GNU GPL.
- *Assumptions concerning decision sequence*: In the initial phase success is hardly foreseeable and it is likely that no decision concerning a license scheme has been made. For successful mature projects other individuals assure the further development.
- *Assumed consequence*: Pro-socially motivated contributors participate mainly in the intermediate phase of a project.
- *Appraisal:* The results are fair. However, pro-social motivation does not seem to increase according to a second order function but logarithmically.

Peer recognition

- *Assumptions concerning motivation*: Developers motivated by peer recognition want to feel comfortable in a community and to interact nicely with each other.
- *Assumptions concerning decision sequence*: At the very beginning of a project there is hardly anybody around to interact with. However, this may change rapidly. Following the team development model of Tuckman it is assumed that in the forming phase it is agreeable to contribute for individuals motivated by peer recognition. With development going on the community is in the storming and norming phase that is less agreeable. In the performing phase being member of community is agreeable again.
- *Assumed consequence*: Developers motivated by peer recognition are not the very first to contribute to a FOSS project because there is hardly anybody around to interact with. However, very soon developers motivated by peer recognition increase their contribution. In the intermediate phase of a project they partly withdraw their engagement. With development going on this engagement increases again.
- *Appraisal:* The empirical results are very satisfactory.

Personal requirement

- *Assumptions concerning motivation*: Developers interested in adjusting software to personal requirements contribute contingent on the value they derive from an application.
- *Assumptions concerning decision sequence*: The more advanced the development of a software application is the more valuable the application is for a user.
- *Assumed consequence*: Developers interested in adjusting software to personal requirements contribute increasingly with project development going on.
- *Appraisal:* The empirical results are only middling. However, this may be attributed to methodological problems.

Reputation investment

- *Motivation*: Investors in reputation are interested in the production of valuable signals. Doing so is in turn related to the visibility and credibility of such signals. However, investors in reputation also take the costs of producing the signal into account.
- *Assumptions concerning decision sequence*: It is thought that visibility and credibility of signals increase during development. At the same time with development going on it is more and more difficult and therefore costly to get code merged.
- *Assumed consequence*: The motivation of reputation investors to contribute to a FOSS project increases with development going on but flattens for mature projects.
- *Appraisal:* The results are fair. However, investment in reputation does not seem to increase logarithmically but linearly.

Investment in human capital

- *Assumptions concerning motivation*: Investors in human capital participate in FOSS projects in order to learn something. There are two different means to do so: learning is enhanced by user or developer (peer review) feedback and by analyzing source code of other contributors.
- *Assumptions concerning decision sequence*: At the very beginning of project development there is hardly feedback. With development going on feedback increases until it gets difficult to produce source code that attracts feedback at all. However, for advanced projects browsing and analyzing the source codes of other developers gets interesting.
- *Assumed consequence*: The possibilities to learn and therefore the motivation related to investment in human capital increase in the initial phase (because feedback increases), decrease in the intermediate phase (because feedback decreases), and increase again (because of source code browsing).
- *Appraisal:* The empirical results are very good. However, the theoretical assumptions are not completely convincing.

The empirical analysis shows that the basic assumption about the relationship between motivation and development phase is, in general, sound. Concerning peculiar types of motivation the empirical results range from very good to poor. Some assumptions need minor to major theoretical adjustments.

It was found theoretically and empirically that motivation as well as governance structures and mechanisms are related to FOSS development phases. Based on the reasoning made in this dissertation it was further argued that interaction effects between motivation and governance structures respectively mechanisms should explain satisfaction with participation in FOSS development. Unfortunately, the interaction hypothesis is not supported by the data. The interaction variables increase the explained variance between 1% and 10% (non-significant). An easy explanation for this non-satisfying result could not be found.

12.1.4 How are governance structures and mechanisms related to the dynamics of collective action?

The question how to motivate employees (e.g. see Herzberg 2003) is probably one of the most important for many commercial firms. In many theories, e.g. Herzberg's two factor theory (Hackman & Oldham 1976: 251), it is assumed that all individuals are similar responsive to the same stimulus (see also Hackman & Oldham 1975). However, the same stimulus does not have the same appeal to different individuals (Hulin & Blood 1968). In fact, this has also been observed in this study (see chapter 11.4.2). For example only investment in reputation but not the other motivation types is positively and significantly correlated to the degree companies contribute currently to a project. All in all this study supports the assumption that a one-size-fits all approach to foster motivation has its flaws.

Certain social dilemmas are solved sequentially (see Yin 1998: 565). In such situations there are different development phases. As argued in this dissertation it is likely that the different types of motivations are not equally distributed over different development phases. In consequence, the selection of the appropriate governance structures and mechanisms must then be adjusted to the development phase. To build up sustainable competitive advantages an organization must adapt dynamically its governance structures and mechanisms contingent on the development phase. It was shown in chapter 7 that this holds true for entire organizations as well as for groups. Organizations and groups must adapt to life cycles. For example in an ***entrepreneurial phase*** of an organization governance structures and mechanisms must enable the flourish of ideas while little planning is needed. If the same organization grows in size and more and more employees are engaged this may get troublesome. Therefore, in a ***formalization and control phase*** the formalization of rules and the institutionalization of procedures are needed.

Certainly, too much "dynamic change" may be bad because contributors may receive the impression of unreliability. Therefore, there is probably a trade-off between adapting to different development phases and consistency. The proper rate of adaption is certainly difficult to locate and more research is needed.

The above presented line of reasoning can also be recognized in the course of action in the Monday demonstrations. There were certainly different phases: For almost a decade there were Monday prayers for peace in the Nikolai Church in Leipzig with a constant number of participants. In autumn 1989 the number of demonstrators increased gradually until early November. Afterwards, the number of demonstrators decreased sharply and oscillated afterwards. This pattern can only be explained by different phases in which:

i. individuals are differently motivated (see below) and in which

ii. the incentives provided by the effective structures and mechanisms change.

The change of structures and mechanisms was partly induced from the outside and partly from the inside of the social movement. There were exogenous events such as the mass migration from the German Democratic Republic to Austria via Czechoslovakia and Hungary, the decision of the SU to shift from the Brezhnev Doctrine to the Sinatra Doctrine and not to interfere in East Germany, and the success of the trade union Solidarity in Poland. These events lessened the sanctions demonstrators had to face. Additionally, by increasing their size the social movement scaled the potential for sanctions even further down. The structures and mechanisms relevant for the demonstrators were before and after November 9, 1989 quite different. The announcement of Günter Schabowski that all traveling restrictions were abolished *immediately* disrupted the incentive to demonstrate for those intending to head for the West completely.

It can be summarized that governance structures and mechanisms can be related to the dynamics of collective action. However, if one compares FOSS development, the Monday demonstrations, and organizational life cycles it seems to be obvious that there is no common pattern for these different occurrences. Therefore, the adaption of governance structures and mechanisms to different development phases must be adjusted for the specific circumstances in question.

12.1.5 How are different types of motivation related to the dynamics of collective action?

It was concluded above that different types of motivation are differently affected by the dynamics of FOSS development. This insight can be complemented by the findings related to the Monday demonstrations. While some demonstrators wanted to leave the German Democratic Republic other demonstrators wanted to change the political system. Additionally, the first few demonstrators had certainly a more intense desire to oppose the regime than individuals who joined the upheaval later. Only by including different types and intensities of motivation the dynamics of collective action in the Monday demonstrations can be explained. This clearly supports the assumptions that different types of motivation are important to explain collective action in general.

The Monday demonstrations are not only an interesting example for itself but also in comparison to FOSS development already discussed in chapter 12.1.2. There is one big similarity and two differences between the Monday demonstrations and FOSS development:

- **Similarity**: Very similar is that individuals have different types of motivation. For the Monday demonstrations this can be recognized by looking at the texts of the chants sung. While some demonstrators shouted „Wir wollen raus!” (We want to get out) others shouted “Wir bleiben hier!” (We stay here). In the terms of the motivation types in this dissertation the former song text may be interpreted as refereeing to a personal requirement while the later may be interpreted as referring to pro-social motivation.
- **Difference**: An interesting difference between the Monday demonstrations and FOSS development is that the change in the motivation types in the course of action is quite diverse. It was concluded that the very first individuals opposing the German Democratic Republic and thereby starting collective action were driven by the desire to live within the truth. It can be argued that this kind of impetus can be called pro-social motivation. Afterwards individuals motivated by personal requirement (“we want to get out”) joined the cause. It can be hypothesized that the sharp decline in demonstrators in early

November was caused by the announcement that all traveling restrictions were abolished. The individuals who wanted to go to the west ("we want to get out") reached their goal and withdraw in consequence from protesting. In contrast, in FOSS development pro-social motivation is not that prominent at the very beginning.

- **Difference**: For FOSS development individual data has been used while in the case of the Monday demonstrations aggregated data has been analyzed. Therefore, we do not know for sure why an individual in the Monday demonstrations joined the cause. However, it is possible to make some inferences about the individuals by referring to the song texts of the chants. Certainly, such an interpretation is somehow subjectively and error-prone. It is not always easy to deduce from aggregated groups to individuals as we know clearly from research about ecological fallacies (see Robinson 1950). Nevertheless, it is thought that such a deduction can be made if the results are interpreted cautiously.

The comparison between the Monday demonstrations and FOSS development has a very important insight: The change in the type of motivation in the dynamics of collective action is not generic but highly dependent on the object in question. It can be concluded that different types of motivation are related to the dynamics of collective action but that there is no fix relationship between different phases in the course of action and certain types of motivation.

The above stated conclusion has far reaching implications. To organize and foster collective action one has to set up proper governance structures and mechanisms. However, which governance structures and mechanisms are proper depends on the type of motivation of individuals. In consequence, one has first to identify how prospective contributors to the collective good are motivated. This is especially important for the early phase of the production of the collective good. It was suggested in chapter 5.7 that for a public good with an S-shaped production function a start-up problem exists. To set a bandwagon in motion it needs individuals who give it a nudge. This initial phase is especially critical because strategic interaction or selective incentives are not always suitable as a motivational force. The first employee who contributes his knowledge to a database, wiki, or an Intranet forum will probably not have much direct benefit from doing so. One cannot be sure whether the database will be a flop and disappear very soon after initialization. However, if an Intranet forum is properly working one may answer a question and hopping that by doing so to get an answer oneself.

It is suggested strongly that all types of motivation are able to solve the start-up problem. While for example *fun seeking* was thought to be important for FOSS development[720] the same motivation is probably not too much important for picketing which is perhaps driven by *peer recognition*. A charity bazaar that has just open may *pay* a celebrity for the visit so that others assess that the bazaar is a place visited by high society.[721] It also may be that the first contributors to a collective good have a vast *personal requirement*. An example may be fishermen catching shrimps at night who help building a lighthouse. *Investors in reputation* may be motivated

[720] In fact, fun seeking proved to be important in the initial phase (see chapter 11.4.6). However, contrary to the corresponding hypothesis the importance of fun seeking did not decline but oscillated in the further development phases.

[721] In some cases it may be possible to pay the individuals who start cooperation or punish those who do not. However, this is certainly not always possible (see also 5.6). *Firstly*, paying cooperators or punishing non-cooperators is sometimes a public good itself. *Secondly*, it is not always possible to identify cooperators and non-cooperators respectively doing so is too costly. *Thirdly*, it may be too costly to pay or punish everybody who cooperates or does not cooperate. *Fourthly*, rewarding and punishment may be imperfect, difficult to control, and it may undermine intrinsic motivation.

to start building a lighthouse if the media is around. *Investors in human capital* may help to start building a lighthouse because they want to learn how to mix mortar.

Which motivation is suitable to solve the start-up problem is contingent on the structure of the good in question and on the circumstances. Certainly, some types of motivation are less likely suitable to start collective action than others. In either way: Individuals who start cooperation respectively give a bandwagon a push are especially motivated. This motivation has to be identified so that the corresponding governance structures and mechanisms can be set up. In consequence, a course to learn to mix mortar has to be organized, a celebrity must be paid to show up, a visit of the media has to be arranged, or the autonomy of individuals has to be ensured.

12.1.6 Is the problem of collective action really shaped by dynamical forces?

Based on the preceding argumentation one may state: Yes, collective action may be shaped by dynamical forces. It is even thought that in many cases dynamical forces are important for collective action. The importance of dynamical forces depends to a high degree on the production function of a collective good (see chapter 5.7). Certainly, dynamical forces are predominately important in step level, S-shape collective goods and collective goods with a decelerating or an accelerating production function. However, these assumptions are not corroborated. Having found two examples, FOSS development and the Monday demonstrations, that are shaped by dynamical forces does not prove that this pattern is universal.

12.2 Governance structures and mechanisms: Lessons for commercial software development

FOSS governance structures and mechanisms have been given much attention in this dissertation. Then the question becomes what can be learned for software development in general. As the examples of Netscape/Mozilla/Firefox (chapter 2.5.3), NetBSD (chapter 4.2.2), and Mambo/Joomla! (chapter 4.3.7) show it is not trivial to set up proper governance structures and mechanisms (see chapter 4.3). However, knowing more about setting up proper governance structures and mechanisms is important because of the success of FOSS. A study concludes that FOSS plays within the DOD a critical role in network security, infrastructure support, software development, and in research (MITRE 2003: 2). Examples of FOSS applications such as Linux, the Apache HTTP server, and Firefox have been briefly described (see chapter 2.2). In terms of market share FOSS applications are first, second, or third-rung products in several markets such as web servers, server operating systems, web browsers, email, and other ICT infrastructure (see chapter 2.10). This urges to research how governance structures and mechanisms contribute to the success of the FOSS phenomenon. Doing so seems to be even more important because Benkler (2006: 3) observes two major moves shifts in most advanced economies:

- The first move is towards an economy centered on *information* such as financial services, software, and science.
- The second move is related to *communication* built on *cheap processes* and *highly interconnected networks*.

Because of these shifts new patterns of production emerges what Benkler calls the networked information economy. In such a networked information economy social exchange in a radically decentralized manner will play a much larger role. Web 2.0 is the corresponding buzzword. It seems to be plausible to argue that the governance structures and mechanisms of FOSS projects are suited for this type of economy. In consequence, if Benkler's observation is correct it may be even more rewarding to learn how FOSS development deals with the demands of a networked information economy. Lessons beyond software development could be learned.

If FOSS is successful and if FOSS development has special governance structures and mechanisms two questions arises:

i. Which of the governance structures and mechanisms contribute to the success of FOSS projects? Certainly, it cannot be taken for granted that all governance structures and mechanisms found in successful FOSS projects are responsible for it. FOSS projects may be successful *despite* certain governance structures and mechanisms.

ii. Which of the FOSS governance structures and mechanisms can be reproduced by commercial software firms to increase their success as well?

In the following, it is *speculated* about these two questions.

12.2.1 Structures and mechanisms responsible for FOSS success

Characteristic governance structures and mechanisms of FOSS development are inter alia i) peer review respectively egoless programming, ii) self-selection as a recruitment principle, iii) computer-mediated interaction, iv) early and often releases, v) user involvement, vi) modularization, vii) decision making structures based on meritocracy, viii) code reuse, and ix) informal coordination mechanisms. The license of a project is also an important characteristic of FOSS projects (see chapter 2.5). It is speculated that four of the above mentioned governance structures and mechanisms are very important for FOSS success:

- *Firstly*, ***peer review*** and egoless programming is thought to be crucial for FOSS development (see chapter 4.3.1). Thereby bugs are found that the initial coder was not aware of.[722] Raymond (1999a: 41) termed the corresponding phrase “given enough eyeballs, all bugs are shallow”. In the reported example of the Therac-25 incident (see chapter 4.3.5) there were no independent software reviews what seems to have been a major problem (Leveson & Turner 1993: 39). This would have been very important because there were complex interactions between various components. In consequence, a report recommended for the future “an independent safety analysis … by a panel (including but not limited to medical physicists)” (Leveson & Turner 1993: 40).

  Analyzing the source code of other programmers does not only improve the source code but also the knowledge of the developers. In chapter 3.2.2.3 it was argued that being able to read and analyze the source code of other programmers is very important for learning. Kernighan and Plauger (1976: book cover) argue that “careful study and imitation of good programs leads to better writing”. The importance of reviewing source code is backed up by the results in this study. Roughly half of all contributors to FOSS projects review and comment source code (see Figure 136).

- *Secondly*, ***code reuse*** is thought to be very important for FOSS success (see chapter 4.3.8). The most obvious examples of code reuse are software libraries that are collections of subprograms. But also modules or components as well as code fragments can be reused. While software reuse is neither

[722] A controversy question is whether peer review benefits or hurts *software security* (Whitlock 2001).

trivial nor without its problems the advantages may be lower development costs, higher productivity, reduced cycle time, lower training costs, easier maintenance, higher quality, lower risks, and better interoperability. If somebody for example wants to create a new web browser he may rely on an existing layout engine (e.g. Gecko). By doing so a developer does not have to develop source code that deals with formatting information (e.g. CSS) and displaying it on the screen. This lowers development costs but also ensures source code quality that a certain developer is probably not able to produce.

- *Thirdly*, it is argued that ***self-selection as a recruitment principle*** respectively the "automatic selection of programmers to work in the areas they know best" (Moody 1997) is another very important factor for FOSS success (see chapter 4.3.2). It is not the projects that find developers but developers who find projects. The effect of self-selection as a recruitment principle is that developers make their contributions in an area that is appropriate to their skill level. Linus Torvalds (2001: 121) argue that the "most effective way to lead is by letting people do things because they want do them, no because you want them to." Self-selection as a recruitment principle reflects the fact that most FOSS project members are not paid for their contributions and that they are therefore not subject to hierarchical subordination.
- *Fourthly*, ***strong copyleft licenses*** are thought to be very important (see chapters 2.5, 2.6.2.1, 4.2.1, and 4.3.9). They ensure legally that FOSS applications remain free but also inform developers what is proper to do and what not.

Clearly, the selection discussed above can (and should) be challenged.

12.2.2 Structures and mechanisms that are reproducible in commercial projects

It is thought that some of the FOSS governance structures and mechanisms are harder to reproduce for commercial firms than others. In the following, the problems related to such a reproduction are discussed.

- ***Peer review*** respectively ***egoless programming***, to ***release early and often,*** and ***modularization*** are probably not harder to implement for commercial firms than for FOSS projects. Agile software development methods such as Extreme Programming (Jeffries 2004) have implemented similar concepts such as pair programming or the accentuation of feedback.
- ***Self-selection as a recruitment principle*** may work excellent for FOSS projects. However, there are certainly limits for this principle in commercial firms. *Firstly*, such a mechanism depends basically on i) a large number of tasks subjects can choose from and ii) a large number of different developers. Otherwise, the chance that a developer finds a task he is really interested in is low. Most commercial firms have a limited number of salaried developers and a limited number of tasks to accomplish. Therefore, the self-selection principle will in most cases not work properly. An interesting variant would be, however, to hire free lancers worldwide. If a task is announced on a worldwide platform the possibility to find a developer who is tremendously interested in the task increases. Certainly, this would cause new problems because valuable knowledge were outsourced. *Secondly*, there is a short supply of intrinsically interesting tasks. If developers are allowed to choose the tasks they are interested in some boring tasks important for customers are perhaps not accomplished. A commercial firm that perhaps suffered from letting to much decision power to developers is Duke Nukem Forever;

the most famous vaporware.[723] Steve Miller from the computer game publisher 3D Realms argues that the tremendously late development schedule of Duke Nukem Forever is due to changes in the workforce, inexperienced developers, and too much ambition (Next-Gen Podcast 2006: 6:57-59:25). Interestingly, he further argues that the management style of the project to leave much to bottom line developers did work well as long as the project was small. However, because the project grew in size a middle management would have been in need. Because it was not implemented the development was delayed. *Thirdly*, related to recruitment is exit. One principal difference between FOSS developers and developers in commercial project is that the exit option respectively stop contributing is easier respectively cheaper for FOSS developers. In commercial firms the exit option for a developer may be very costly because he his the breadwinner. In some cases he will not find a new job or an equally paid job. In FOSS development an unhappy developer can leave without having to accept a monetary loss. As shown in chapter 5.4.2.1.1 the possibility of an exit option influences behavior in social dilemmas. There are two opposite implications based on the costs of the exit option:

i) Commercial software developers are easier to manage because they will use the threat of exit less frequently. Using the exit option and leaving the firm involves the loss of the salary and is therefore in some cases costly.

ii) Commercial software developers are harder to manage because they will hide their true level of motivation in order not to be laid off (see also chapter 5.6 on the problem of finding intrinsically motivated individuals).

It can be concluded that concerning self-selection not only recruitment but also exit is import. For the self-selection principle to work properly individuals do not only need to join a project because it is fun but also to leave it if the fun has disappeared. Doing so may be, however, very costly for developers in commercial projects.

A practical solution for the implementation of the self-selection principle is Google's "20 percent time" rule discussed in chapters 3.2.1 and 5.6.

- ***Computer-mediated interaction*** is certainly nothing special anymore. There seem to be no reasons why commercial software development could not raise the level of computer-mediated interaction to the level of FOSS projects.
- ***User involvement*** is probably difficult to foster for commercial firms because people do like to donate scare resources to a public good but not to commercial firms. A developer may do not mind to work hundreds of hours to implement a new feature if the corresponding application is licensed under a FOSS license (see also Franck & Jungwirth 2003a: 7 and 11). However, if a commercial firm appropriates it and makes a lot of money out of it this may be different. Hansmann (1980) coined the term non-distribution constraint for describing this phenomenon. The solution for this problem is to put the project under a FOSS license. While doing so may be problematic in some cases it is not in other cases.

[723] Vaporware is software that is announced in advance of its release but fails to emerge. Duke Nukem Forever has been developed for more than ten years. Other examples of vaporware are the GNU Hurd kernel or the Project Xanadu in development since about 1960 (Wolf 1995).

- While some of the technical problems related to ***code reuse*** do not seem to be more or less severe for commercial projects than for FOSS projects there is a major obstacle. The source code to be reused must fit the requirements. Therefore, the source code repository to choose from must have a certain size. Otherwise, the demanded source code will hardly be found in the repository. Certainly, this does not hold true if the project is an upgrade of old software.

  There are probably examples of big software houses like Microsoft, Oracle, or Adobe Systems with an adequate source code repository. However, smaller software companies probably lack a source code repository beyond the projects they have already completed in-house. In FLOSS development this problems does not exist for projects with a GNU GPL compatible license.[724] The permission to run, study, modify, and distribute the source code provides a very good condition for the reuse of source code across different and non-related projects. That is why FOSS development is perfectly suited for source code reuse. The transaction costs for code reuse are very low compared to a source code exchange between commercial firms. Concerning transaction costs Coase (1960: 15) writes (see also Niehans 2003; Picot & Dietl 1990: 178):

  > *In order to carry out a market transaction it is necessary to discover who it is that one wishes to deal with, to inform people that one wishes to deal and on what terms, to conduct negotiations leading up to a bargain, to draw up the contract, to undertake the inspection needed to make sure that the terms of the contract are being observed, and so on. These operations are often extremely costly, sufficiently costly at any rate to prevent many transactions that would be carried out in a world in which the pricing system worked without cost.*

  The search costs are lower for FOSS than for commercial projects because FOSS projects do not hide their source code. Commercial firms have in contrast an interest to do so. For source code under a FOSS license there are no bargaining costs. The license terms allow reusing the code. In contrast, in the case of commercial firms the terms of transaction have to be negotiated. As experience from software requirement specification suggests this is very complicated (McConnell 1996: 137). The approach to specify all requirements of an application at the beginning of a project as suggested in the waterfall development model has failed. For Haefliger et al. (2007: 27) there may be two additional reasons why source code reuses works especially well in FOSS development. *Firstly*, developers' mobility across software projects leads to reuse of source code from parallel or subsequent projects. [725] *Secondly*, the reputation rewards from peers for clean, modular, and well-structured source code provide incentives to write reusable components.

  Raymond (2004: 381) argued that talking about source code reuse without talking about FOSS is hardly possible. Haefliger et al. (2007: 18) found that in FOSS development "developers routinely and widely reuse software components".[726] Therefore, the solution for firms that want to take advantage of

---

[724] Certainly, this does not hold true for strange FOSS licenses incompatible with the GNU GPL.

[725] Haefliger et al. (2007: 27) do not restrict their argument concerning developer mobility and code reuse to FOSS projects. However, it seems to be that code reuse based on developer mobility in the case of commercial projects is partly constrained by intellectual property rights. Certainly, this does not apply for developer mobility within a legal entity.

[726] The authors identified 55 reused components comprising 2,975 reuse incidents (Haefliger, von Krogh, & Spaeth 2007: 25).

the power of source code reuse is straightforward: The own software has to be put under a sound FOSS license. Thereby the huge source code repository of other FOSS projects can be reused.

- ***Informal coordination mechanisms*** are certainly harder to grasp or judge by outsiders respectively by the management than formal coordination mechanisms. A manager who intends to measure project progress will have a harder time if coordination mechanisms are informal. That is perhaps why the management of software projects value formal coordination mechanisms highly. Kraut and Streeter (1995: 80) conclude that:

  > *senior managers are likely to be major beneficiaries of formal management procedures. These procedures are often used to enable senior managers to establish control and gain feedback about the software development process. … Software metric data and status reviews were used {in the company the scientist studied} primarily by senior managers and had little impact on the day-to-day software development process.*

  At the same time empirical studies (Curtis, Krasner, & Iscoe 1988: 1278-1282; Herbsleb & Grinter 1999: 2; Perry, Staudenmayer, & Votta 1994: 42-43) suggest that not only developers in FOSS projects but also developers in commercial software projects rely heavily on informal coordination mechanisms.

  For community driven FOSS projects there is no need to give outsiders an account. Developers have no responsibilities to the outside. Therefore, no control from the outside is needed. An application is released if it is ready. While this FOSS development style may cause sometimes problems it also increases the quality of source code. Commercial firms are, in contrast, accountable to shareholders and different stakeholders. To rely only on informal coordination processes may cause some problems related to the ability for outsiders to control software developers. The Duke Nukem Forever example discussed above is possibly a commercial software project that suffered from insufficient control from the outside. To rely only on informal coordination mechanisms may be therefore difficult to reproduce for commercial firms because there are responsibilities to the outside.

- ***Decision making structures based on meritocracy*** suffer probably from similar problems already discussed for self-selection as a recruitment principle and informal coordination. Projects dominated by high-rated developers may lack services demanded by customers but not valued by developers. Additionally, the management may have a control problem in projects dominated by developer meritocracy.

- Whether putting source code under a ***strong copyleft license*** is difficult for a commercial firm or not depends on the distribution of copyrights and the structure of the source code in question. If the copyrights are exclusively hold by the commercial firm no major practical problems should arise. However, problems arise if the software incorporates source code of third parties (see e.g. Netscape Navigator or Java). Finding this source code, asking the third parties for a relicense permission, or rewrite such code from scratch may be troublesome. Certainly, the related problems depend on the structure of the source code and the importance of the code in question. It is probably easier to unhinge source code of third parties if the source code is strictly modularized than if it is spaghetti code.

Conclusion: What commercial firms can learn from FOSS development

It was guessed that peer review, code reuse, self-selection as a recruitment principle, and the GNU GPL license are the most crucial governance structures and mechanisms responsible for FOSS success. It was also suggested that some FOSS governance structures and mechanisms are harder to reproduce than others are. It is thought that self-selection as a recruitment principle, user involvement, informal coordination mechanisms, and decision making based on meritocracy are hard to reproduce for commercial firms. Code reuse is insofar hard to reproduce for commercial firms because the corresponding transaction costs are high and because their own source code repository is low. It is argued that modularization, to release early and often, peer review, and computer-mediated interaction are not harder to maintain for commercial projects than for FOSS projects.

Merging the argumentation above it can be concluded that for commercial firms that intend to reproduce FOSS governance structures and mechanisms peer review may be especially promising because it i) seems to be an important aspect of FOSS development and ii) relatively easy to reproduce by commercial firms. Code reuse, self-selection as a recruitment principle, and putting the source code under the GNU GPL license seems to be more difficult to reproduce.

The discussion above shows additionally that commercial firms that intend to push the use of FOSS governance structures and mechanisms further will hardly get around putting their own source code under a FOSS license. Otherwise, commercials firms cannot take full advantage of user involvement and source code reuse. If source code is released under a FOSS license there is a strong rational to use the GNU GPL or at least a GNU GPL compatible license.

The comments made above should clearly not be interpreted as a recommendation to put software applications under a FOSS license respectively under the GNU GPL. It is simply a recommendation to put an application under a FOSS license ***if*** one wants to make use of FOSS governance structures and mechanisms. To answer the question whether it is beneficial to adopt FOSS governance structures and mechanisms is far beyond this dissertation.

12.2.3 FOSS: A silver bullet for software development?

In a highly influential article Fred Brooks concluded that there is no silver bullet, no single technology, or no management technique that solve all problems in software development (Brooks 1987: 10). Following an argument of Aristotle he argues that there will be no such silver bullet because difficulties respectively *essential complexity* is rooted in the very nature of software development (Brooks 1987: 11). Only *accidental complexity* is not inherent in software development and can therefore be avoided. Accidental complexity happens because of inefficient feature implementation.[727] Advancement in technology and development methods such as high-level languages has addressed such accidental complexity. The speed of development advancement slows down because more and more of accidental complexity is mastered. Programmers spend more time on essential complexity what can hardly be accelerated with technology and development methods.

Glass (2005) argues in contrast to Brooks that there were and will always be new milestones in software history. In consequence, accidental complexity does not fade away. Anyway: there is no silver bullet for

[727] In an attempt to expand the categorization of Brooks (1987: 11) Raymond (2004: 299-302) introduces optional complexity which is tied to a desirable feature. Optional complexity can simply be eliminated by changing the objectives of a project.

software development; either because of essential complexity or because of accidental complexity. In consequence, it is clear that FOSS development is not a silver bullet (see Ambler 2007). This remark is made because sometimes the FOSS development model is advertised as a silver bullet for software development. While FOSS development and its mechanisms and structures are certainly thrilling they will not solve all problems of software development.

12.3 User integration: Some remarks for commercial projects

This dissertation highlighted that users might contribute to the production of goods or services. To rely on users for production sounds convenient but leads to a major difficulty. Individuals must be motivated to contribute. For commercial firms the obvious and in many cases very sound solution to this problem is simply paying users for their contributions. It is for example possible to include every user who gives feedback in a lottery. However, as already described this is not without its problem (see chapter 5.6). Doing so is costly, rewarding may be imperfect, multi-tasking problems may arise, intrinsic motivation may be undermined, problems of performance control arise, and contributions collapse if payment is withdrawn.

An alternative to paying users is to rely on other types of motivation. Users may just have fun to contribute, may feel an inner obligation to do so, may be motivated because of peer recognition, may want to learn something, may have a special personal requirement, or may want to produce a valuable signal. Obviously, relying on such motivation is not without its pitfalls too. Tasks that are related to fun, an inner obligation, peer recognition, human capital, personal requirement, and investment in reputation may be in short supply. Correspondingly, there may be not enough individuals that have such a motivation. Even if there are enough individuals with such types of motivation it may be not trivial to find them. However, the most difficult problem seems to be to match organizational needs and individual interests. The desires of users and commercial firms may diverge. While for example users may demand that the fruits of their contributions are available to other users for free this may be not in the interest of commercial firms.

In either case: the involvement of users in the development of products or services demands prominently that their motivation is taken into account. While doing so is certainly also wise within commercial firms it grows in importance if external users are incorporated in the production process.

12.4 Further research

History of social science as well as common sense informs us that no single explanation succeeds in the end (see also Tilly 1984: 80-143).[728] While for example some concepts or theories are good in explaining social change others explain social stability more successfully.[729] In consequence, research that uses different concepts and theories are needed to amplify the here presented results.

[728] This may even be true for natural science. According to Morris (1991: 17-28) there are at least nine different competing theories why zebras have stripes. Some theories argue that the stripes are a camouflage mechanism while others claim that it is an optical illusion for confusing lions or the tsetse fly. Other theories assume that the stripes are a mean for identification, a mean for social identity, or a thermoregulatory mechanism. All theories are plausible but all have some flaws. Therefore, it is not surprising that science came not to a final conclusion why zebras have stripes.

[729] Certainly, this dissertation focused primarily on the dynamics and not on the stability of collective action in FOSS development. It would be also interesting to learn more about the stability of collective action in FOSS development.

Fisher (1971: 16) argued that a hypothesis “is never proved or established, but is possibly disproved, in the course of experimentation. Every experiment may be said to exist only in order to give the facts a chance of disproving the null hypothesis”. If hypotheses cannot be rejected they are more likely to be true than before the analysis was executed (Popper 1984: 198). A theory gets more and more corroborated by failed attempts of falsification. Therefore research focused on the same subject is needed to corroborate the here presented results.

In the following possible questions for further research are discussed. These research questions are either focused on the hypotheses researched in this dissertation but also on completely different aspects that emerged in the course of this dissertation.

12.4.1 Development phases

Corroboration of the principal idea

It is thought that the principal idea that the motivation of developers is contingent on the development phase is supported. Certainly, this impression has to be corroborated by additional studies. To do so different methodological approaches would be most suitable.

Adjustment of single assumptions: Motivation

No theoretical adjustments for the relationship between peer recognition and personal requirement on the one hand and development phases on the other hand seem to be indicated. The corresponding assumptions related to investment in reputation, pro-social motivation, and investment in human capital need some adjustment and refinement. Corresponding further research is needed. Most interestingly would be to research the relationship between fun seeking and development phases. For this relationship the theory must be completely revised.

Adjustment of single assumptions: Governance structures and mechanisms

The question of autonomy was thought to be an important factor influencing fun seeking. It was also argued that autonomy decreases during development. However, no such relationship was found. It would be interesting to inquire whether this result is due to the used methodology, the used sampling approach, or to theoretical wrong assumptions. All of these explanations seem to be possible. Researching autonomy seems to be of major interest because of its importance for fun seeking and because fun seeking is in turn a very important type of motivation in FOSS development.

Organizational change

This study has shown how the composition of motivation may change during FOSS development. As argued in this dissertation different types of motivation have different needs. In consequence, the governance structures and mechanisms must be adapted to the changing composition of motivation. This leads to the question how FOSS projects adapt themselves to the changing demands respectively how it is possible to manage organizational change. The following questions seem to be relevant:

- How does one know in which phase the project is? Answering this question seems to be practically very difficult. Development is most likely non-linear and there will not be a clear cut between two different phases. The discussion of business cycles (Kuznets 1940; Schumpeter 1939a, 1939b) showed that even if it is known what to do in a specific phase of a cycle it is no easy to determine the phase of a cycle.
- If it is known what to do in a certain phase and if one also knows in which phase of development a project is one still has to solve the problem of change. As we know from change management literature (e.g. Kotter 1996) this may not be an easy task.

Longitudinal study

For testing the main hypothesis of this dissertation a truly longitudinal study on FOSS projects is needed. Thereby it could be distinguished between period, age, and cohort effects (see chapter 9.1).

A longitudinal study *about FOSS projects* could be accompanied by a longitudinal study *about FOSS developers*. This would allow answering the question whether a different composition of motivation in the different development phases is due to motivational shifts within developers or due to a shift in the composition of differently motivated individuals.

12.4.2 Probability sample

A very important task is to conduct a large-scale FOSS study based on a *probability sample*. This study could serve then as a reference for other studies. However, drawing a proper probability sample seems to be very difficult in the realm of FOSS development. In consequence drawing a proper probability sample has not even approximately accomplished.

12.4.3 The relative decline of the GNU GPL

In chapter 2.5.2 a rather surprising fact was presented. At SourceForge the relative share of the GNU GPL decreased from 2001 to 2006 in relation to other FOSS licenses. It is highly relevant and interesting to research this observation. The following research questions could be analyzed:

i. *Firstly*, is the observed phenomenon restricted to SourceForge?
ii. *Secondly*, is the decline in the share based on deliberate decisions of developers against the GNU GPL or is it just an artifact? It may for example be that abandoned projects licensed under the GNU GPL are more strictly removed from SourceForge by the project members than projects under other licenses. The relative decline of the GNU GPL is then artificial and not very relevant. It also may be that GNU GPL projects have more frequently moved to other hosting platforms or that such projects decided more frequently to install their own infrastructure.
iii. *Thirdly*, if the decline is based on a deliberate decision against the GNU GPL one may analyze the corresponding reasons.
iv. *Fourthly*, if the decline in shares is not an artifact one may search for the long-term consequences.

12.4.4 Cultural influences on participation in FOSS projects

A striking pattern in all studies on FOSS developers is the low participation rate of Asian developers. There is no paper known that addresses this puzzle. From the very preliminary study presented in chapter 2.4 some hypothesis could be generated. It was found that the participation rate in FOSS development depends i) on the number of inhabitants of a country and ii) more interestingly on the individualism of a country. Language does not seem to play a significant role for participation in FOSS development. To inquire whether these preliminary results endure a theoretically and methodologically more sophisticated study would be appealing.

12.4.5 Researching Conway's law

In chapter 4.3.9 Conway's law has been presented. Basically, it states that the structure of an organization determines the corresponding structure of the source code. It would be interesting to analyze empirically whether or how the composition of developers in a FOSS project does influence the structure of the source code. For example it could be argued that the number of developers and their dominant type of motivation is related to modularization.

In chapter 4.3.9 it was also mentioned that in some cases product modularity influences organizational modularity. Based on this observation it would be appealing to research whether the type of software application influences the organizational structure of FOSS projects. In consequence, it could be researched why and to what extent the source code influences the structure of a project.

12.4.6 Different roles in FOSS development

Another rather surprising result of this study is that a considerable fraction of FOSS project members is not involved in programming (see chapter 10.3.3). Roughly one-fifth of subjects neither develop features nor fix bugs (see chapter 11.2.3). A subject of this study noted:

> *It is not only programmers in a community for a program or OS. … There are quite some people who test or document and yet can be considered hackers. … Are they the subject of the next PhD?*

Indeed, further research may analyze the involvement of subjects who are not involved in programming. Answering related question is interesting because of two different reasons:

i. *Firstly*, the success of most software applications is probably predominantly based on its source code. However, other factors such as documentation, localization, or help for users contribute as well to the success of an application. Researching subjects contributing not source code but other services helps then to understand the success of FOSS better.

ii. *Secondly*, in chapter 4.4 it was suggested that FOSS governance structures and mechanism are not completely unique to software development. In chapter 4.5 the example of pirate democracy was discussed to highlight the power of self-governance. Researching FOSS development beyond producing source code helps to understand which of the governance structures and mechanisms can be reproduced in areas beyond programming.

12.4.7 Decision making structure and leadership

In chapter 4.5 the peculiar excurse of pirate democracy was discussed. It was highlighted how important democratic decision making procedures and voting were in pirate communities. These procedures seem to have influenced the motivation of the pirates considerably. Because of their higher motivation to fight they were superior to the crew of merchant ships that were directed in a very hierarchical manner. Based on this example one may argue that a democratic decision making structure is always beneficial if motivation beyond carrot and stick is important.

Interestingly, it was found in chapter 4.3.7 that there are two different types of FOSS projects: In the first type a benevolent dictator rules the project (e.g. Linux kernel) while the second type is based on democratic decision making (e.g. Apache Software Foundation). There are very successful examples of both types of projects. Therefore, it cannot be argued that projects ruled by a benevolent dictator are inferior (or superior) to democratic decision making. Certainly, both types of projects heavily rest on meritocracy and developers may leave the project whenever they want. Nevertheless, the basic questions remain:

i. Under which circumstances do projects adopt the one or the other decision making process? It may be that the degree of heterogeneity of developers is a factor influencing the decision making process (see Ellickson 1993: 1349; Hansmann 1990: 1783). One would expect an elevated degree of democratic decision making if the FOSS developers in a project have a high degree of homogeneity.

ii. What are the consequences of different decision making styles? It may be that the decision making style influences the social atmosphere (see White & Lippitt 1960). Based on the findings of White and Lippitt one may argue that benevolent dictators produce more critical discontent and aggressive behavior while democratic decision making produces more friendly conversation, high work motivation, and originality.

There are no theoretical or empirical studies related to the question of decision making in FOSS projects. In hypotheses 5 to 8 (see chapter 8.2.2.1) it was attempted to contribute to this question. However, the corresponding results (see chapter 11.4.2.1) were not insightful at all. It is argued here that to research different decision making and leadership styles in FOSS is one of the very most interesting questions in FOSS research.

12.4.8 Comparison to social movements

To compare social movements (see e.g. chapter 6.4) with FOSS communities would be an interesting topic. While there are certainly similarities such as informal organization and the involvement of “ordinary people” (Tilly 2004: 3) there also differences.[730] For such a comparison the study of Hertel et al. (2003) provides an exciting source of inspiration.

12.5 Welcome to the conversation

In a foreword of a book on FOSS development Eric Raymond (2002: xi) writes that the content of the book is not the last word because last words are about dead things. FOSS is quite lively and its development model

[730] For a multifaceted overview over social movements see Snow et al. (2004).

will evolve and change its parameters. There are therefore no last words concerning FOSS. This also applies to this analysis. There are obviously mistakes and inaccuracies in this dissertation. In consequence, further research, opposition, and different opinions are needed to scope with the ongoing change, to scope with aspects not researched yet, and to scope with inaccuracies and mistakes of this dissertation. To support related efforts this dissertation (including anonymized data) is put under strong copyleft licenses that *allow modification legally*. To *allow modification practically* this dissertation is published in a wiki (http://www.bernhard-kuster.org/wiki) at least for one year after official publication. Readers are welcome to change and supplement content. As Eric Raymond (2002: xi) puts it: "***Welcome to the conversation***".

Appendix I: Survey[731]

Free and Open Source Software Survey

PhD project of Bernhard Kuster

[The layout of this survey has been supplemented with the seal of the University of Zurich and with approximately 200 logos of different FOSS projects. Due to the fact that the author is neither the owner of the corresponding intellectual property rights nor that rights are granted to put the emblems under a Creative Commons respectively a GNU FDL license these emblems have been removed.]

[731] The questions in the distributed survey were not numbered.

[University seal]

University of Zurich
Institute for Organization and Administrative Science

Free and **O**pen **S**ource **S**oftware Survey

Goals of our study
We are interested in the reasons why hackers are involved in FOSS (Free and Open Source Software) projects, mainly how different project structures influence their motivation to contribute to them. Why do we conduct a survey? Our goal is to get developers own opinion on a range of issues that have not been covered by scientific research yet.

The PhD thesis of Bernhard Kuster is based on the data collected by this survey.

Refer your answer to…
If applicable, please refer your answers to *the FOSS project you spent most time on during the last year*.

Time need
Answering the survey takes about 15 minutes.

Returning the questionnaire
There is a box next to the main entrance where you can hand in your completed survey. It is also possible to post the completed survey to: University of Zurich, Institute for Organization and Administrative Science, Bernhard Kuster, Plattenstrasse 14, CH-8032 Zurich, Switzerland.

Who we are
We belong to a research group at the University of Zurich (Switzerland) which focuses on free and open source software research. If you have any questions, please contact Bernhard Kuster in person next to the main entrance or by email (bernhard.kuster@iou.unizh.ch).

Results
The results of this study will be published online (http://www.kuster.be/results) as soon as the project is completed. The results will be published under a *GNU Free Documentation License / Creative Commons* (attribution / share-alike) double license. Your data will be kept strictly confidential. Results will only be published in an anonymous form.

[University seal]

University of Zurich
Institute for Organization and Administrative Science

In the questions below we will always use the term FOSS (Free and Open Source Software), although we know that many developers make a sharp distinction between these two concepts.

1 Do you think of yourself as a part of the Free Software or as a part of the Open Source community?

Free Software community....... ☐ Open Source community...... ☐ I do not care..................... ☐ I do not know.................. ☐

If applicable, please refer from now on all your answer to the project you spent most time on during the last year.

	The following questions concern <u>the FOSS project itself.</u>	**strongly disagree**				**strongly agree**
2	It is pretty easy for an outsider to get code into the project.	☐	☐	☐	☐	☐
3	Looking back, it was a good decision to participate in this project.	☐	☐	☐	☐	☐
4	The code review process is strongly influenced by some sort of voting.	☐	☐	☐	☐	☐
5	Controversial decisions are explained to the ones not involved in decision making.	☐	☐	☐	☐	☐
6	Companies made important contributions to the project <u>at the beginning</u>.	☐	☐	☐	☐	☐
7	Companies made important contributions to the project <u>recently</u>.	☐	☐	☐	☐	☐
8	In this project I get interesting comments on my submitted code.	☐	☐	☐	☐	☐
9	If this project were to disappear it would be missed by ... the entire FOSS community.	☐	☐	☐	☐	☐
10	... project developers.	☐	☐	☐	☐	☐
11	... its users.	☐	☐	☐	☐	☐

12 How are formal and informal project positions (e.g. release manager, project leader) appointed? *(Select the most accurate answer)*

Somehow elected by the community.................................. ☐ Appointed by programmer(s) holding a special position......... ☐

Chosen by informal consent .... ☐ Project is too small for special positions .... ☐

13 Please specify the project software. *(Select the most accurate answer)*

Kernel/Drivers................................. ☐ Server software.............................. ☐ Development tools/Libraries .... ☐

Graphics/Audio/Multimedia.............. ☐ Games... ☐ Office applications/Productivity tools .. ☐

Other, ____

14 Approximately, how many programmers are contributing actively source code currently? ≈ __________ programmers

[University seal]

University of Zurich
Institute for Organization and Administrative Science

15 Which of the following project tools are used for development? *(Check all that apply)*

Versioning system ☐ Bug tracking system ☐ Mailing lists ☐

Automated regression testing ☐ IRC ☐ Other, ____________________

16 Please indicate the phase of development *(Select the most accurate answer)*

Planning ☐ Proof of concept / for developers ☐ Basic feature complete ☐

Ready for end-user ☐ Feature complete / mature ☐

17 How does your project credit to contributors? *(Check all that apply)*

Website ☐ Credit file ☐ Write access source repository ☐

Other, ____________________________________

18 What is (are) the license(s) of this project? (*List all licenses. If the licensing model is not yet decided, please fill in "not decided")*

1st 2nd 3rd ____________________

19 What do you think about this license model?

I prefer another license model, namely:

I am happy with this license model ☐ I can not decide ☐

20 Is your software included in the latest official release of at least one of the major Linux distributions?
(e.g. Debian, Fedora, Gentoo, Knoppix, Ubuntu, RedHat Enterprise Linux, Slackware, SUSE, other)

Yes ☐ No ☐ I don't know / not Linux compatible ☐

The following questions concern <u>your engagement</u> in the FOSS project.

21 Which roles do you hold for your project? (Check all that apply)

Adding or developing features ☐ Fix bugs ☐ Report bugs ☐

Review / comment source code ☐ Create graphics ☐ Write How-To's or FAQ ☐

Translate (e.g. interfaces, FAQ) ☐ Marketing / promoting ☐ Donate ☐

Help users / answer questions ☐ Project management ☐ Other, ____________________

22 In which year did you start contributing to this project? In __________

23 How many hours per week did you work on this project in the last year? On the average __________hours per week.

24 If you are paid for project participation: How many hours a week are you paid for? __________hours Not paid ☐

[University seal]

University of Zurich
Institute for Organization and Administrative Science

		strongly decrease				strongly increase
25	What do you think: will your level of contribution to this project increase or decrease?	☐	☐	☐	☐	☐

	The following questions refer to <u>your attitudes towards the FOSS project</u>.	**strongly disagree**				**strongly agree**
26	Programming in this project is fun. I really enjoy it.	☐	☐	☐	☐	☐
27	For me personally it is important that my source code is elegant.	☐	☐	☐	☐	☐
28	Because of my involvement in this FOSS project, I hope to get a better job.	☐	☐	☐	☐	☐
29	I learn more from contributing to this project than from taking a programming course.	☐	☐	☐	☐	☐
30	FOSS development is one of the big innovations of the 20th century.	☐	☐	☐	☐	☐
31	I would prefer not giving back source code to the project but license terms force it.	☐	☐	☐	☐	☐
32	For me it is important that my project peers value what I am doing.	☐	☐	☐	☐	☐
33	Project participation gives me confidence that I would be able to fork the project if something went wrong.	☐	☐	☐	☐	☐
34	The satisfaction of seeing the resulting code is very important.	☐	☐	☐	☐	☐
35	I would feel ashamed if other community members thought that my code was bad.	☐	☐	☐	☐	☐
36	Companies are interested in employing talented project members.	☐	☐	☐	☐	☐
37	My participation in this FOSS project ensures that the software provides functionality that matches my personal needs.	☐	☐	☐	☐	☐
38	I do not mind using proprietary software if it serves me better.	☐	☐	☐	☐	☐
39	Giving public credit is important so that non-community members can judge my skills.	☐	☐	☐	☐	☐
40	Participating in this FOSS project sharpened my programming skills.	☐	☐	☐	☐	☐
41	For me it is important that my peers give me useful feedback on my source code.	☐	☐	☐	☐	☐
42	Being able to fix problems that personally bother me is one of the key advantages of my involvement in this FOSS project.	☐	☐	☐	☐	☐
43	I make concessions about my source code to please other project members even if I am not convinced that it enhances code quality.	☐	☐	☐	☐	☐
44	Access to source code is an ethical problem similar to free speech.	☐	☐	☐	☐	☐

[University seal] University of Zurich
Institute for Organization and Administrative Science

The questions beneath refer to the relationship between your participation in the FOSS project and business interests.

45 Is there a relationship between your participation in this FOSS project and business interests?

Yes, I am paid for my project participation.. ☐ Yes, my own business is related to my project participation .... ☐ There is no relationship...................... ☐

Please skip the next four questions if there is no relationship between your project participation and business interests.

	My project participation ensures that…	strongly disagree				strongly agree
46	… the software provides demanded functionalities.	☐	☐	☐	☐	☐
47	… bugs get fixed.	☐	☐	☐	☐	☐
48	… there is the know - how to fork the project if anything goes wrong.	☐	☐	☐	☐	☐
49	… a good reputation in the FOSS community is maintained or built up.	☐	☐	☐	☐	☐

	The following questions are adapted to FOSS from standard questionnaires and describe different aspects of work situations.	strongly disagree				strongly agree
50	I feel like I have a lot of influence whether my patches get merged.	☐	☐	☐	☐	☐
51	I feel pressured to use the given project tools (e.g. bug tracking system or CVS).	☐	☐	☐	☐	☐
52	People in the project tell me I am good at what I do.	☐	☐	☐	☐	☐
53	I am free to express my ideas and opinions on the project.	☐	☐	☐	☐	☐
54	I keep my communication with other contributors minimal.	☐	☐	☐	☐	☐
55	Most days I feel a sense of accomplishment from programming.	☐	☐	☐	☐	☐
56	In case I was to leave the project people would miss me even if there was a good replacement for me.	☐	☐	☐	☐	☐
57	I really like the people in this project.	☐	☐	☐	☐	☐
58	In the project I do not get much of a chance to show how capable I am.	☐	☐	☐	☐	☐

As usual, at the end of the questionnaire we also need some demographic information. Please be assured that these data will also be kept strictly confidential.

59 In which year were you born? In __59a__ In which year did you start programming? In __59b__

60 How many hours per week do you spend working in a paid job and commuting to that job? ___________ hours

[University seal]

University of Zurich
Institute for Organization and Administrative Science

61 What is the highest level of education you have completed?

Compulsory education ☐ | Qualification above compulsory education but no entry requirement for universities ☐ | Entry requirement for universities ☐
University: Bachelor ☐ | University: Master ☐ | University: PhD ☐

62 Are you a student? Yes ☐ No ☐

63 Is your occupation or are your studies somehow related to IT?

Yes, related to software ☐ Yes, related to hardware ☐ No ☐ Don't work/study presently ☐

64 If your occupation or your studies is / are related to IT, please specify. *(Leave otherwise blank)*

Development ☐ Support ☐ System administration ☐
Sales / marketing / consulting ☐ Student ☐ Professor/lecturer/assistant/teacher ☐

65 Overall, how good is your English? **not very good** ☐ ☐ ☐ ☐ ☐ **very good**

66 What is your monthly gross income? (before taxes or insurance fees are taken away / 1 Dollar ≈ 0.8 Euro)

< 500 US Dollar ☐ 501 - 2000 US Dollar ☐ 2001 - 4000 US Dollar ☐
4001 - 7000 US Dollar ☐ > 7000 US Dollar ☐ I prefer not answering this question ☐

67 I would like to get an email with the results. My email is: ______________________________

68 If you like to comment on motivation for participation in FOSS projects you can note down your annotations here:

Thank you for your participation!

Free and **O**pen **S**ource **S**oftware Survey

PhD project of Bernhard Kuster

[Logos of different FOSS projects]

Appendix II: Additional results of the survey analysis

			Fun seeking		Reputation		Human capital			Pro social				Peer recognition		Personal requirement		
			A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
Fun seeking	Code elegance	A		0.26 ***	0.04	0.09	0.18 **	0.14 *	0.16 **	0.15 *	0.13 *	0.19 **	0.15 **	0.32 ***	0.21 ***	0.00	0.16 **	0.30 ***
	Resulting code	B	0.26 ***		0.12	-0.01	0.25 ***	0.24 ***	0.08	-0.05	-0.04	0.31 ***	-0.09	0.23 **	0.25 ***	0.30 ***	0.17 **	0.29 ***
Reputation	Better job	C	0.04	0.12		0.39 ***	0.22 ***	-0.11	0.19 **	-0.17 **	-0.04	0.14 *	-0.09	-0.05	0.12	0.14 *	-0.07	0.10
	Talented programmers	D	0.09	-0.01	0.39 ***		0.20 **	-0.09	0.06	-0.03	-0.06	0.09	0.02	-0.01	0.22 ***	0.05	0.08	0.07
Human capital	Programming skills	E	0.18 **	0.25 ***	0.22 ***	0.20 **		0.21 ***	0.38 ***	0.00	-0.04	0.20 ***	0.17 **	0.00	0.17 **	0.05	0.00	0.16 **
	Feedback importance	F	0.14 *	0.24 ***	-0.11	-0.09	0.21 ***		0.33 ***	0.00	0.04	0.15 *	0.08	0.01	0.22 ***	0.04	0.07	0.14 *
	Better than course	G	0.16 **	0.08	0.19 **	0.06	0.38 ***	0.33 ***		0.17 **	0.00	0.15 *	0.09	0.01	0.08	0.06	0.12	0.13 *
Pro social	Proprietary software	H	0.15 *	-0.05	-0.17 **	-0.03	0.00	0.00	0.17 **		0.28 ***	0.06	0.21 ***	0.02	0.05	-0.03	0.05	-0.04
	Ethical problem	I	0.13 *	-0.04	-0.04	-0.06	-0.04	0.04	0.00	0.28 ***		0.24 ***	0.17 **	0.22 **	0.04	-0.06	0.11	0.00
	Innovation	J	0.19 **	0.31 ***	0.14 *	0.09	0.20 ***	0.15 *	0.15 *	0.06	0.24 ***		0.09	0.23 **	0.22 ***	0.17 **	0.20 **	0.30 ***
	License force	K	0.15 **	-0.09	-0.09	0.02	0.17 **	0.08	0.09	0.21 ***	0.17 **	0.09		-0.04	0.05	-0.16 **	0.04	0.02
Peer recognition	Bad Code	L	0.32 ***	0.23 **	-0.05	-0.01	0.00	0.01	0.01	0.02	0.22 **	0.23 **	-0.04		0.21 **	-0.04	0.31 ***	0.15
	Peer appraisal	M	0.21 ***	0.25 ***	0.12	0.22 **	0.17 **	0.22 ***	0.08	0.05	0.04	0.22 ***	0.05	0.21 **		0.13 *	0.19 **	0.12
Personal requirement	Fork	N	0.00	0.30 ***	0.14 *	0.05	0.05	0.04	0.06	-0.03	-0.06	0.17 **	-0.16 **	-0.04	0.13 *		0.18 **	0.20 **
	Personal needs	O	0.16 **	0.17 **	-0.07	0.08	0.00	0.07	0.12	0.05	0.11	0.20 **	0.04	0.31 ***	0.19 **	0.18 **		0.45 ***
	Fix problems	P	0.30 ***	0.29 ***	0.10	0.07	0.16 **	0.14	0.13 *	-0.04	0.00	0.30 ***	0.02	0.15	0.12	0.20 **	0.45 ***	

* Significant at the 0.1 level / ** Significant at the 0.05 level / *** Significant at the 0.01 level

Figure 203: Correlation of items included in motivation scales

Appendix III: Interaction effects and satisfaction with FOSS development

Satisfaction with FOSS particpation: Fun seeking

Model	Item	B	Std. Error	Beta	t	Significance
1	Constant	4.383	0.44		9.971	0
	Gross income	0.141	0.098	0.224	1.44	0.154
	Study	0.168	0.198	0.116	0.847	0.399
	Education	-0.036	0.069	-0.063	-0.517	0.606
	Hours working	-0.004	0.005	-0.113	-0.775	0.44
	Programming expericence	-0.008	0.02	-0.049	-0.426	0.671
	Age	-0.003	0.013	-0.037	-0.277	0.783
2	Constant	4.337	0.714		6.073	0
	Gross income	0.141	0.099	0.224	1.432	0.156
	Study	0.168	0.199	0.116	0.845	0.401
	Education	-0.036	0.069	-0.063	-0.515	0.608
	Hours working	-0.004	0.005	-0.114	-0.775	0.44
	Programming expericence	-0.008	0.02	-0.047	-0.392	0.696
	Age	-0.003	0.013	-0.037	-0.275	0.784
	Fun seeking	0.01	0.116	0.009	0.082	0.935
3	Constant	3.161	0.904		3.497	0.001
	Gross income	0.125	0.097	0.199	1.29	0.201
	Study	0.163	0.197	0.112	0.826	0.411
	Education	-0.058	0.069	-0.103	-0.852	0.397
	Hours working	-0.003	0.005	-0.079	-0.55	0.584
	Programming expericence	-0.008	0.02	-0.044	-0.374	0.71
	Age	-0.002	0.012	-0.024	-0.184	0.854
	Fun seeking	-0.064	0.118	-0.061	-0.54	0.591
	Feedback	-0.064	0.077	-0.093	-0.828	0.41
	Missed by developers	0.144	0.068	0.235	2.133	0.036
	Autonomy	0.213	0.124	0.207	1.713	0.091
	Competence	0.014	0.123	0.014	0.112	0.911
	Relatedness	0.073	0.102	0.084	0.708	0.481
4	Constant	-3.119	4.228		-0.738	0.463
	Gross income	0.174	0.099	0.277	1.761	0.082
	Study	0.118	0.202	0.081	0.584	0.561
	Education	-0.098	0.072	-0.174	-1.362	0.177
	Hours working	-0.001	0.005	-0.028	-0.197	0.844
	Programming expericence	-0.003	0.02	-0.017	-0.142	0.888
	Age	-0.004	0.013	-0.04	-0.279	0.781
	Fun seeking	1.407	0.959	1.342	1.467	0.147
	Feedback	0.582	0.55	0.85	1.058	0.293
	Missed by developers	1.244	0.524	2.02	2.376	0.02
	Autonomy	0.432	1.021	0.419	0.423	0.674
	Competence	0.324	0.9	0.321	0.36	0.72
	Relatedness	-0.505	0.801	-0.582	-0.631	0.53
	Interaction effect fun x autonomy	-0.063	0.237	-0.385	-0.266	0.791
	Interaction effect fun x competence	-0.077	0.206	-0.501	-0.376	0.708
	Interaction effect fun x relatedness	0.141	0.187	0.915	0.753	0.454
	Interaction effect fun x feedback	-0.143	0.126	-1.071	-1.139	0.258
	Interaction effect fun x missed by developers	-0.248	0.118	-2.107	-2.109	0.038

Figure 204: Interaction effects and satisfaction with FOSS development: Complete data fun seeking

Satisfaction with FOSS particpation: Pro social motivation						
Model	**R^2**	**Change in R^2**	**Change in F**	**df1[a]**	**df2[b]**	**Significance in change of F**
1 Demographics	0.058	0.058	0.680	6	66	0.666
2 Demographics, Motivation	0.108	0.050	3.611	1	65	0.062
3 Demographics, motivation, structures and mechanisms	0.151	0.043	0.774	4	61	0.546
4 Demographics, motivation, structures and mechanisms, interaction effects	0.230	0.079	1.470	4	57	0.223

[a] Nominator degress of freedom [b] Denominator degress of freedom

Figure 205: Interaction effects and satisfaction with FOSS development: Pro-social motivation[732]

Satisfaction with FOSS particpation: Peer recognition						
Model	**R^2**	**Change in R^2**	**Change in F**	**df1[a]**	**df2[b]**	**Significance in change of F**
1 Demographics	0.264	0.264	2.097	6	35	0.079
2 Demographics, Motivation	0.267	0.003	0.142	1	34	0.709
3 Demographics, motivation, structures and mechanisms	0.329	0.062	0.693	4	30	0.603
4 Demographics, motivation, structures and mechanisms, interaction effects	0.427	0.098	0.851	5	25	0.527

[a] Nominator degress of freedom [b] Denominator degress of freedom

Figure 206: Interaction effects and satisfaction with FOSS development: Peer recognition[733]

[732] Demographic items included are gross income, student, education, working hours, programming experience, and age. The motivation item is pro-social motivation. Items related to structures and mechanisms are missed by community, missed by users, happiness with license, and copyleft license. The interaction items are pro-social motivation_x_missed by community, pro-social motivation_x_missed by users, pro-social motivation_x_happiness with license, and pro-social motivation_x_copyleft license.

[733] Demographic items included are gross income, student, education, working hours, programming experience, and age. The motivation item is peer recognition. Items related to structures and mechanisms are relatedness, missed by community, missed by developers, and number of programmers. The interaction items are peer recognition_x_relatedness, peer recognition_x_missed by community, peer recognition_x_missed by developers, and peer recognition_x_number of programmers.

Satisfaction with FOSS particpation: Human capital						
Model	R^2	Change in R^2	Change in F	df1[a]	df2[b]	Significance in change of F
1 Demographics	0.057	0.057	0.812	6	81	0.564
2 Demographics, Motivation	0.134	0.078	7.170	1	80	0.009
3 Demographics, motivation, structures and mechanisms	0.240	0.106	2.639	4	76	0.040
4 Demographics, motivation, structures and mechanisms, interaction effects	0.246	0.006	0.142	4	72	0.966

[a] Nominator degress of freedom [b] Denominator degress of freedom

Figure 207: Interaction effects and satisfaction with FOSS development: Human capital[734]

Satisfaction with FOSS particpation: Reputation investment						
Model	R^2	Change in R^2	Change in F	df1[a]	df2[b]	Significance in change of F
1 Demographics	0.165	0.165	2.345	6	71	0.040
2 Demographics, Motivation	0.170	0.005	0.394	1	70	0.532
3 Demographics, motivation, structures and mechanisms	0.313	0.143	1.870	7	63	0.090
4 Demographics, motivation, structures and mechanisms, interaction effects	0.326	0.013	0.218	5	58	0.953

[a] Nominator degress of freedom [b] Denominator degress of freedom

Figure 208: Interaction effects and satisfaction with FOSS development: Reputation investment[735]

[734] Demographic items included are gross income, student, education, working hours, programming experience, and age. The motivation item is human capital. Items related to structures and mechanisms are missed by developers, decision explained, feedback, and competence. The interaction items are human capital_x_missed by developers, human capital_x_decision explained, human capital_x_feedback, and human capital_x_competence.

[735] Demographic items included are gross income, student, education, working hours, programming experience, and age. The motivation item is investment in reputation. Items related to structures and mechanisms are missed by community, missed by users, company contribution recently, number of programmers, number of modes to credit, and credit mechanisms (dummy yes/no). The interaction items are investment in reputation_x_missed by community, investment in reputation_x_missed by users, investment in reputation_x_company contribution recently, investment in reputation_x_number of programmers, investment in reputation_x_number of modes to credit, and investment in reputation_x_credit mechanisms (dummy yes/no).

Satisfaction with FOSS particpation: Personal requriement						
Model	**R^2**	**Change in R^2**	**Change in F**	**df1[a]**	**df2[b]**	**Significance in change of F**
1 Demographics	0.082	0.082	1.084	6	73	0.380
2 Demographics, Motivation	0.089	0.007	0.547	1	72	0.462
3 Demographics, motivation, structures and mechanisms	0.112	0.024	0.933	2	70	0.398
4 Demographics, motivation, structures and mechanisms, interaction effects	0.121	0.009	0.342	2	68	0.711

[a] Nominator degress of freedom [b] Denominator degress of freedom

Figure 209: Interaction effects and satisfaction with FOSS development: Personal requirement[736]

[736] Demographic items included are gross income, student, education, working hours, programming experience, and age. The motivation item is personal requirement. Items related to structures and mechanisms are missed by users and included in distribution. The interaction items are personal requirement_x_missed by users and personal requirement_x_included in distribution.

Appendix IV: Hypotheses overview

Y = Yes, hypothesis is supported

N = No, hypothesis is not supported

P = Hypothesis is partly supported

#	Hypotheses: Motivation	Supported
1	Fun seeking and investment in human capital are the two most important types of motivation.	Y
2	Reputation investment is the least important motivation.	Y
3a	Personal requirement is positively related to fun seeking.	Y
3b	Personal requirement is positively related to investment in human capital.	Y
4	Fun seeking, pro-social motivation, and peer recognition group while personal requirement, personal requirement, investment in reputation, and investment in human capital group as well.	Y

#	Hypotheses: Motivation and governance structures respectively mechanisms	Supported
5	The fraction of projects that have no special mechanisms for assigning project positions is considerably lower for very small projects.	Y
6	Fun seeking is positively related to projects in which project positions are assigned by informal consent.	N
7	Peer recognition is negatively related to projects in which project positions are assigned by appointment.	N
8	Reputation investment is positively related to projects in which positions are assigned by elections.	N
9a	Fun seeking is positively related to autonomy.	N
9b	Fun seeking is positively related to competence.	Y
9c	Fun seeking is positively related to relatedness.	N
9d	Fun seeking is positively related to positive feedback.	Y
10	Fun seeking is positively related to projects that would be missed by developers if they were to disappear.	Y
11	The intensity of pro-social motivation is positively related to happiness with the license of a project.	Y
12	The intensity of pro-social motivation is positively related to projects with strong copyleft licenses.	Y
13a	Pro-social motivation is positively related to projects that would be missed by the users if they were to disappear.	N
13b	Pro-social motivation is positively related to projects that would be missed by the entire community if they were to disappear.	N
14a	Peer recognition is positively related to relatedness.	Y
14b	Peer recognition is positively related to the number of programmers contributing to a project.	N
15a	Peer recognition is positively related to projects that would be missed by its developers if they were to disappear.	N
15b	Peer recognition is positively related to projects that would be missed by the entire community if they were to disappear.	N

16	Personal requirement is positively related to projects that would be missed by the users if they were to disappear.	Y
17	Investment in personal requirement is positively related to projects that are included in a major Linux distribution.	N
18a	Reputation investment is positively related to the degree a project would be missed by the entire community if it were to disappear.	Y
18b	Reputation investment is positively related to the degree a project would be missed by the users if it were to disappear.	Y
19a	Reputation investment is positively related to the existence of deliberately created credit mechanisms.	N
19b	Reputation investment is positively related to the number of different means to credit contributions.	Y
20	Reputation investment is positively related to the involvement of commercial firms.	Y
21	Reputation investment is positively related to the number of developers in a project.	Y
22a	Investment in human capital is positively related to feedback giving.	Y
22b	Investment in human capital is positively related to the degree decisions are explained.	N
23	Investment in human capital is positively related to competence.	Y
24	Investment in human capital is positively related to projects that would be missed by the developers if they were to disappear.	Y
25	Paid developers are more likely engaged in projects to which commercial firms contribute as well.	Y
26a	Fun seeking is negatively related to the status of being paid.	N
26b	The motivational types related to external regulation are not affected by the status of being paid.	N
27	The development phase is positively related to the involvement of commercial firms.	Y
28	Paid developers are more likely engaged in projects that are in a later development phase.	Y

#	**Hypotheses: Interaction of motivation and governances structures/mechanisms**	**Supported**
29	After having controlled for demographics, motivation, and governance structures and mechanisms interaction effects between motivation and governance structures and mechanisms should explain satisfaction with participation in FOSS development.	N

#	**Hypotheses: Motivation and the type of engagement**	**Supported**
30a	Fun seeking is positively related to adding or developing features.	Y
30b	Fun seeking is positively related to reviewing source code.	Y
31a	Pro-social motivation is positively related to translating.	Y
31b	Pro-social motivation is positively related to marketing and promoting.	N
31c	Pro-social motivation is positively related to creating graphics.	Y
31d	Pro-social motivation is positively related to writing FAQ's.	Y
31e	Pro-social motivation is positively related to donating.	Y
32a	Peer recognition is positively related to reviewing source code.	N
32b	Peer recognition is positively related to helping users.	N
33a	Personal requirement is positively related to reporting bugs.	N
33b	Personal requirement is positively related to fixing bugs.	N
33c	Personal requirement is positively related to adding or developing features.	Y
34a	Reputation investment is positively related to adding or developing features.	N
34b	Reputation investment is positively related to promoting and marketing.	Y

35a	Investment in human capital is positively related to adding or developing features.	N
35b	Investment in human capital is positively related to bug fixing.	N
35c	Investment in human capital is positively related to reviewing source code.	Y

#	**Hypotheses: Motivation and community affiliation**	**Supported**
36a	Pro-social motivation is positively related to being part of the free software and negatively related to being part of the open source community.	Y
36b	Other types of motivation are not related to community affiliation.	Y
37	Being part of the free software community is positively related to contribute to projects with a strong copyleft licenses.	Y
38	Being part of the free software community is negatively related to business interests.	Y

#	**Hypotheses: Motivation and development phases**	**Supported**
39a	Motivation increases or decreases in different development phases contingent on the type of motivation.	Y
39b	The distribution of motivation in the development phases can be described best by the specified functions.	P

#	**Hypotheses: Development phases and governance structures respectively mechanisms**	**Supported**
40	During project development the fraction of projects included in a major Linux distribution increases.	P
41	During project development it gets increasingly difficult for an outsider to get code merged.	Y
42	During project development the number of developers increases.	Y
43	During project development the involvement of commercial firms increases.	Y
44	During project development projects are increasingly missed by the entire community if they were to disappear.	N
45	During project development the project is increasingly missed by the users if it were to disappear.	Y
46	The number of different means to credit increases with project development.	Y
47	During project development autonomy decreases.	N
48	During project development project are decreasingly missed by the developers if they were to disappear.	N
49	In the initial development phase the fraction of projects that have chosen a license increases.	N
50	In the initial development phase the fraction of projects that have some sort of credit mechanism increases.	N
51	During project development feedback giving increases in the initial phase, decreases in the intermediate phase, and increases in the mature phase.	N
52	During project development explaining decisions increases in the initial phase, decreases in the intermediate phase, and increases in the mature phase.	N

Appendix V: GNU Free Documentation License

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this

License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (c) YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this: with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST. If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Bibliography

Aaker, D.A., & R.P. Bagozzi. (1979). Unobservable variables in structural equation models with an application in industrial selling. *Journal of Marketing Research, 16*(2), 147-158.

AAPOR. (2006). *Standard definitions: Final dispositions of case codes and outcome rates for surveys* (4 ed.). Lenexa: AAPOR.

Abrams, B.A., & M.D. Schmitz. (1984). The crowding-out effect of governmental transfers on private charitable contributions: Cross section evidence. *National Tax Journal, 37*(4), 563-568.

Abrams, B.A., & M.D. Schmitz. (1985). The crowding-out effect of governmental transfers: A rejoinder. *National Tax Journal, 38*(4), 575-576.

Acheson, M. (1987). The lobster fiefs revisited: Economic and ecological effects of territoriality in the Maine lobster industry. In: B. J. McCay & J. M. Acheson (Eds.), *The Question of the commons: The culture and ecology of communal resources* (pp. 37-65). Tucson: University of Arizona Press.

Adams, J.S. (1963). Toward an understanding of inequity. *Journal of Abnormal and Social Psychology, 67*(5), 422-436.

Adizes, I. (1979). Organizational passages: Diagnosing and treating lifecycle problems of organizations. *Organizational Dynamics, 8*(1), 3-25.

Adrianson, L., & E. Hjelmquist. (1999). Group processes in solving two problems: face-to-face and computer-mediated communication. *Behaviour & Information Technology, 18*(3), 179-198.

AEA. (2006). *American Economic Association application for year 2007*. Retrieved 14. December, 2006, from https://www.lbmchost.com/aea/Membership.asp.

Aguinis, H., C.A. Henle, & C. Ostroff. (2001). Measurement in work and organizational psychology. In: N. Anderson, D. S. Ones, H. K. Sinangil & C. Viswesvaran (Eds.), *Handbook of industrial, work and organizational psychology* (Vol. 1, pp. 10-26). London: Sage.

Aiken, L.R. (1985). *Psychological testing and assessment* (5th. ed.). Boston: Allyn & Bacon.

Akaike, H. (1973). *Information theory as an extension of the maximum likelihood principle.* Paper presented at the Second International Symposium on Information Theory, Budapest.

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control, 19*(6), 716-723.

Akaike, H. (1981). Citation Classic: A new look at the statistical-model identification. *Current Contents/Engineering Technology & Applied Sciences, 51*, 42.

Akerlof, G.A. (1980). A theory of social custom, of which unemployment may be one consequence. *Quarterly Journal of Economics, 94*(4), 749-775.

Albers, S., & L. Hildbrandt. (2006). Methodische Probleme bei der Erfolgsfaktorenforschung - Messfehler, formative versus reflektive Indikatoren und die Wahl des Strukturgleichungs - Modells. *ZfbF, 58*(2), 2-33.

Albert, H. (1985). *Treatise on critical reason.* Princeton: Princeton University Press.

Albert, H. (1991). *Traktat über kritische Vernunft* (5 ed.). Tübingen: Mohr.

Alchian, A.A., & H. Demsetz. (1972). Production, information costs, and economic organization. *American Economic Review, 62*(5), 777-795.

Aldisert, R.J. (1997). *Logic for lawyers: A guide to clear legal thinking* (3 ed.). South Bend: National Institute for Trial Advocacy.

Aldrich, H.E. (1972). Technology and organizational structure: Reexamination of findings of the Aston Group. *Administrative Science Quarterly, 17*(1), 26-43.

Alexa. (2007). *About the Alexa traffic rankings: Some important disclaimers* (16. March). Retrieved 16. March, 2007, from http://www.alexa.com/site/help/traffic_learn_more.

Alexander, C.S., & H.J. Becker. (1978). Use of vignettes in survey-research. *Public Opinion Quarterly, 42*(1), 93-104.

Allais, M. (1953). Le comportement de l'homme rationnel devant le risque: Critique des postulats et axiomes de l'ecole Americaine. *Econometrica, 21*(4), 503-546.

Allen, R.C. (1983). Collective invention. *Journal of Economic Behavior & Organization, 4*(1), 1-24.

Allison, G.T., & P. Zelikow. (1999). *Essence of decision: Explaining the Cuban Missile Crisis* (2 ed.). New York: Longman.

Amabile, T.M. (1996). *Creativity in context update to the social psychology of creativity*. Boulder: Westview Press.

Amabile, T.M. (1997). Motivating creativity in organizations: On doing what you love and loving what you do. *California Management Review, 40*(1), 39-58.

Amabile, T.M., W. DeJong, & M.R. Lepper. (1976). Effects of externally imposed deadlines on subsequent intrinsic motivation. *Journal of Personality and Social Psychology, 34*(1), 92-98.

Ambler, S.W. (2007). The quest for the silver bullet: Looking for the elusive solution. *Dr Dobbs Journal, 32*(4), 99-101.

American Psychological Association, American Educational Research Association, National Council on Measurement in Education, & American Psychological Association. (1985). *Standards for educational and psychological testing*. Washington, DC: American Psychological Association.

Amor-Iglesias, J.J., J.M. González-Barahona, G. Robles-Martínez, & I. Herráiz-Tabernero. (2005). Measuring libre software using Debian 3.1 (Sarge) as a case study: Preliminary results. *Upgrade: The European Journal for the Informatics Professional, 6*(3), 13-16.

Anderson, C., & N.R. Franks. (2001). Teams in animal societies. *Behavioral Ecology, 12*(5), 534-540.

Anderson, J.C., & D.W. Gerbing. (1982). Some methods for respecifying measurement models to obtain unidimensional construct measurement. *Journal of Marketing Research, 19*(4), 453-460.

Anderson, L.R., & C.A. Holt. (1997). Information cascades in the laboratory. *American Economic Review, 87*(5), 847-862.

Andreoni, J. (1995a). Cooperation in public goods experiments: Kindness or confusion? *American Economic Review, 85*(4), 891-904.

Andreoni, J. (1995b). Warm-glow versus cold-prickle: The effects of positive and negative framing on cooperation in experiments. *Quarterly Journal of Economics, 110*(1), 1-21.

Annabi, H., K. Crowston, & J. Howison. (2003, 14.-17. December). *Defining open source software project success.* Paper presented at the International Conference on Information Systems, Seattle.

Anonymous. (2006). An interstate process perspective on political gerrymandering. *Harvard Law Review, 119*(5), 1576-1597.

Anonymous. (2008). *"I am rich" guided tour*. YouTube (8. August). Retrieved 14. August, 2008, from http://www.youtube.com/watch?v=noZcgSTmDA4.

Ante, S.E., J. Greene, S. Hamm, & J. Kerstetter. (2003). The Linux uprising. *Business Week*(3822), 78-84.

Apache Software Foundation. (2006a). *Apache HTTP Server version 2.2: Module index*. Retrieved 13. September, 2006, from http://httpd.apache.org/docs/2.2/mod/.

Apache Software Foundation. (2006b). *Consensus gauging through voting*. Retrieved 11. July, 2006, from http://www.apache.org/foundation/voting.html.

Apache Software Foundation. (2006c). *How the ASF works*. Retrieved 21. September, 2006, from http://apache.org/foundation/how-it-works.html.

Apple. (2006). *Kernel programming guide: Glossary - Apple Public Source License* (7. November). Retrieved 6. February, 2007, from http://developer.apple.com/documentation/Darwin/Conceptual/KernelProgramming/Glossary/chapter_21_section_1.html.

Apweiler, R., A. Bairoch, C.H. Wu, W.C. Barker, B. Boeckmann, S. Ferro, et al. (2004). UniProt: The Universal Protein knowledgebase. *Nucleic Acids Research, 32*(Database issue), D115-D119.

Arazy, O., W. Morgan, & R. Patterson. (2006). *Wisdom of the crowds: Decentralized knowledge construction in Wikipedia.* Paper presented at the 16th Workshop on Information Technologies and Systems, Milwaukee.

Aris, J. (2000). Inventing systems engineering. *IEEE Annals of the History of Computing, 22*(3), 4-15.

Aristoteles. (1994). *Metaphysik* (H. Bonitz, Trans.). Reinbek bei Hamburg: Rowohlt.

Aristotle. (1912). *A Treatise on Government* (W. Ellis, Trans.). London: J. M. Dent & Sons.

Aristotle. (1998). *The Nicomachean ethics* (W. D. Ross, J. L. Ackrill & J. O. Urmson, Trans.). Oxford: Oxford University Press.

Arnold, H.J., & D.C. Feldman. (1981). Social desirability response bias in self-report choice situations. *Academy of Management Journal, 24*(2), 377-385.

Arrow, K.J. (1950). A difficulty in the concept of social welfare. *Journal of Political Economy, 58*(4), 328-346.

Arrow, K.J. (1972). Gifts and exchanges. *Philosophy & Public Affairs, 1*(4), 343-362.

Arrow, K.J. (1974). *The limits of oganization*. New York: W.W. Norton.

Arthur, W.B. (1989). Competing technologies, increasing returns, and lock-in by historical events. *Economic Journal, 99*(394), 116-131.

Asay, M. (2005). *Open-source licensing: GPL is a better model* (07. April). Retrieved 12. September, 2006, from http://www.networkworld.com/supp/2005/opensource/070405-face-off-no.html.

Aumann, R.J. (2005). *War and peace*. Nobelprize.org (8. December). Retrieved 20. June, 2007, from http://nobelprize.org/nobel_prizes/economics/laureates/2005/aumann-lecture.pdf.

Avis, N.E., K.W. Smith, C.L. Link, G.N. Hortobagyi, & E. Rivera. (2006). Factors associated with participation in breast cancer treatment clinical trials. *Journal of Clinical Oncology, 24*(12), 1860-1867.

Axelrod, R. (1984). *The evolution of cooperation*. New York: Basic Books.

Axelrod, R., & W.D. Hamilton. (1981). The evolution of cooperation. *Science, 211*(4489), 1390-1396.

Baard, P.P., E.L. Deci, & R.M. Ryan. (2004). Intrinsic need satisfaction: A motivational basis of performance and well-being in two work settings. *Journal of Applied Social Psychology, 34*(10), 2045-2068.

Babcock, C. (2005). *Sun jumps on open source database bandwagon to boost Solaris* Information Week. Retrieved 19. Mai, 2006, from http://www.informationweek.com/story/showArticle.jhtml?articleID=174301092.

Baden, J. (1998). A new primer for the management of common resources and public goods. In: J. Baden & D. S. Noonan (Eds.), *Managing the commons* (2 ed., pp. 51-62). Bloomington: Indiana University Press.

Bagnoli, M., & M. McKee. (1991). Voluntary contribution games: Efficient private provision of public goods. *Economic Inquiry, 29*(2), 351-366.

Bailey, K.D. (1994). *Methods of social research* (4 ed.). New York: Free Press

Baker, G.P., M.C. Jensen, & K.J. Murphy. (1988). Compensation and incentives: Practice vs. theory. *Journal of Finance, 43*(3), 593-616.

Baldwin, C.Y., & K.B. Clark. (2000). *Design rules* (Vol. 1). Cambridge: MIT Press.

Bales, R.F. (1950). A set of categories for the analysis of small group interaction. *American Sociological Review, 15*(2), 257-263.

Bales, R.F., & F.L. Strodtbeck. (1960). Phases in group problem solving. In: D. Cartwright & A. F. Zander (Eds.), *Group dynamics: Research and theory* (2 ed., pp. 624-638). Evanston: Row Peterson & Company.

Ball, P. (2007). *The more, the wikier: The secret to the quality of Wikipedia entries is lots of edits by lots of people*. news@nature.com (27. February). Retrieved 2. March, 2007, from http://www.nature.com/news/2007/070226/full/070226-6.html.

Banerjee, A.V. (1992). A simple-model of herd behavior. *Quarterly Journal of Economics, 107*(3), 797-817.

Bank of Japan. (2006). *Financial markets: Foreign exchange market*. Retrieved 6. September, 2006, from http://www.boj.or.jp/en/type/stat/dlong/fin_stat/rate/#forex.

Banker, R.D., J.M. Field, R.G. Schroeder, & K.K. Sinha. (1996). Impact of work teams on manufacturing performance: A longitudinal field study. *Academy of Management Journal, 39*(4), 867-890.

Barnard, C.I. (1938). *The functions of the executive*. Cambridge: Harvard University Press.

Barney, J. (1991). Firm resources and sustainable competitive advantage. *Journal of Management, 17*(1), 99-120.

Baron, J., & J. Jurney. (1993). Norms against voting for coerced reform. *Journal of Personality and Social Psychology, 64*(3), 347-355.

Baron, R.M., & D.A. Kenny. (1986). The moderator-mediator variable distinction in social psychological research: Conceptual, strategic, and statistical considerations. *Journal of Personality and Social Psychology, 51*(6), 1173-1182.

Baroudi, J.J., & W.J. Orlikowski. (1989). The problem of statistical power in MIS research. *MIS Quarterly, 13*(1), 87-106.

Barretthoward, E., & T.R. Tyler. (1986). Procedural justice as a criterion in allocation decisions. *Journal of Personality and Social Psychology, 50*(2), 296-304.

Barry, B., & R.L. Oliver. (1996). Affect in dyadic negotiation: A model and propositions. *Organizational Behavior and Human Decision Processes, 67*(2), 127-143.

Bartels, L.M. (1987). Candidate choice and the dynamics of the presidential nominating process. *American Journal of Political Science, 31*(1), 1-30.

Bartlett, C.A., & S. Ghoshal. (1996). Beyond strategy, structure, systems to purpose, process, people: Reflections on a voyage of discovery. In: P. B. Duffy (Ed.), *The relevance of a decade essays to mark the first ten years of the Harvard Business School Press* (pp. 323-345). Boston: Harvard Business School Press.

Batson, C.D., & L.L. Shaw. (1991). Evidence for altruism: Toward a pluralism of prosocial motives. *Psychological Inquiry, 2*(2), 107-122.

Battelle, J., & E. Schmidt. (2005). *The 70 percent solution*. CNNMoney.com (28. November). Retrieved 22. June, 2007, from http://money.cnn.com/2005/11/28/news/newsmakers/schmidt_biz20_1205/.

Baumeister, R.F., & M.R. Leary. (1995). The need to belong: Desire for interpersonal attachments as a fundamental human motivation. *Psychological Bulletin, 117*(3), 497-529.

BBC News. (2003). *Cold War hotline recalled*. Retrieved 23. November, 2006, from http://news.bbc.co.uk/2/hi/europe/2971558.stm.

BBC News. (2005). *Can sound really travel 200 miles?* (13. December). Retrieved 31. Mai, 200, from http://news.bbc.co.uk/1/hi/magazine/4521232.stm.

BBC News. (2006a). *1985: Miners jailed for pit strike murder*. Retrieved 19. December, 2006, from http://news.bbc.co.uk/onthisday/hi/dates/stories/may/16/newsid_2512000/2512469.stm.

BBC News. (2006b). *Man turns paper clip into house* (11. July). Retrieved 8. August, 2006, from http://news.bbc.co.uk/2/hi/technology/5167388.stm.

Beck, K., M. Beedle, A.v. Bennekum, A. Cockburn, W. Cunningham, M. Fowler, et al. (2001). *Principles behind the Agile Manifesto*. Retrieved 18. January, 2007, from http://agilemanifesto.org/principles.html.

Becker, G.S. (1962). Investment in human capital: A theoretical analysis. *Journal of Political Economy, 70*(5), 9-49.

Becker, G.S. (1991). A note on restaurant pricing and other examples of social influences on price. *Journal of Political Economy, 99*(5), 1109-1116.

Becker, G.S. (1996). *Accounting for tastes*. Cambridge: Harvard University Press.

Becker, G.S. (1997). The economic way of looking at life. In: T. Persson (Ed.), *Nobel lectures in economic sciences (1991-1995)* (pp. 38-58). Singapore: World Scientific.

Bedell, J., & F. Sistrunk. (1973). Power, opportunity costs, and sex in a mixed-motive game. *Journal of Personality and Social Psychology, 25*(2), 219-226.

Behlendorf, B. (1999). Open source as a business strategy. In: C. DiBona, S. Ockman & M. Stone (Eds.), *Open sources: Voices from the open source revolution* (1 ed., pp. 149-170). Sebastopol: O'Reilly.

Benabou, R., & J. Tirole. (2005). Incentives and prosocial behavior. *NBER Working Paper Series, 11535*, 1-43.

Benedict, R. (1989). *The chrysanthemum and the sword: Patterns of Japanese culture*. Boston: Houghton Mifflin.

Benkler, Y. (2002). Coase's penguin, or, Linux and the nature of the firm. *Yale Law Journal, 112*(3), 369-446.

Benkler, Y. (2006). *The wealth of networks: How social production transforms markets and freedom*. New Haven: Yale University Press.

Bennet, K. (2003). Linux GPL and binary module exception clause? *Linux kernel mailing list, 3. December 2003*.

Benston, G.J., M. Bromwich, & A. Wagenhofer. (2006). Principles- versus rules-based accounting standards: The FASB's standard setting strategy. *Abacus-a Journal of Accounting Finance and Business Studies, 42*(2), 165-188.

Bentham, J. (1789). *An introduction to the principles of morals and legislation*. London: printed for T. Payne & Son.

Benz, M., & S. Meier. (2005). Do people behave in experiments as in real life? Evidence from donations. *Working Paper Institute for Empirical Research in Economics, University of Zurich, 248*, 1-18.

Berglund, E., & M. Priestly. (2001). Open source documentation: In search of user driven, just-in-time writing, *International Conference on Computer Documentation* (pp. 132-141). Santa Fe: ACM Press.

Bergstrom, T.C. (1970). Scandinavian consensus solution for efficient income distribution among nonmalevolent consumers. *Journal of Economic Theory, 2*(4), 383-398.

Berk, R.A. (1974). Gaming approach to crowd behavior. *American Sociological Review, 39*(3), 355-373.

BerliOS. (2006). *Software map: Development status* (22. September). Retrieved 22. September, 2006, from http://www.developer.berlios.de/softwaremap/trove_list.php?form_cat=2

Bessen, J., & R.M. Hunt. (2004). An empirical look at software patents. *Federal Reserve Bank of Philadelphia Working Paper, 03-17*(R), 1-22.

Bessen, J., & E. Maskin. (2002). Sequential innovation, patents, and imitation. *Working Paper of the School of Social Science at the Institute for Advanced Study, 25*(updated 2006), 1-55.

Bettenhausen, K., & K.J. Murnighan. (1985). The emergence of norms in competitive decision making groups. *Administrative Science Quarterly, 30*, 350-372.

Betz, D., & J. Edwards. (1986). *BYTE interview with Richard Stallman*. Free Software Foundation (July). Retrieved 12. February, 2007, from http://www.gnu.org/gnu/byte-interview.html.

Beverland, M., & L.S. Lockshin. (2001). Organizational life cycles in small New Zealand wineries. *Journal of Small Business Management, 39*(4), 354-362.

Bezroukov, N. (1999a). Open source software development as a special type of academic research (Critique of vulgar raymondism). *First Monday, 4*(10).

Bezroukov, N. (1999b). A second look at the cathedral and the bazaar. *First Monday, 4*(12).

Bezroukov, N. (2004, 02 April 2004). *Portraits of open source pioneers: Benevolent dictator - A slightly skeptical unauthorized biography and the first ten years of Linux*. Retrieved 18. Mai, 2004, from http://www.softpanorama.org/People/Torvalds/index.shtml.

Biancuzzi, F. (2006). *Confessions of a recovering NetBSD zealot*. ONLamp.com. Retrieved 18. September, 2006, from http://www.onlamp.com/pub/a/bsd/2006/09/14/netbsd_future.html?page=1.

Bickel, P.J., E.A. Hammel, & J.W. Oconnell. (1975). Sex bias in graduate admissions: Data from Berkeley. *Science, 187*(4175), 398-404.

Biel, A. (2000). Factors promoting cooperation in the laboratory, in common pool resource dilemmas, and in large-scale dilemmas: Similarities and differences. In: M. Van Vugt (Ed.), *Cooperation in modern society : promoting the welfare of communities, states, and organizations* (Vol. 1, pp. 25-42). London: Routledge.

Biel, A., D. Eek, & T. Gärling. (1997). Distributive justice and willingness to pay for municipality child care. *Social Justice Research, 10*, 63-80.

Bierstedt, R. (1963). *The social order: An introduction to sociology* (2 ed.). New York: McGraw-Hill.

Bies, R.J., & J.S. Moag. (1986). Interactional justice: Communication criteria of fairness. In: R. J. Lewicki, B. H. Sheppard & B. H. Bazerman (Eds.), *Research on negotiations in organizations* (Vol. 1, pp. 43-55). Greenwich CT.

Bies, R.J., & T.M. Tripp. (2001). A passion for justice: The rationality and morality of revenge. In: R. Cropanzano (Ed.), *Justice in the workplace: From theory to practice* (pp. 197-226). Mahwah: Lawrence Erlbaum Associates.

Bikhchandani, S., D. Hirshleifer, & I. Welch. (1992). A theory of fads, fashion, custom, and cultural change as informational cascades. *Journal of Political Economy, 100*(5), 992-1026.

Blackler, F. (1995). Knowledge, knowledge work and organizations: An overview and interpretation. *Organization Studies, 16*(6), 1021-1046.

Blair, T. (2006). *I will become the patron of a London community sports club. I will work with the club over the years as the Olympics approaches in 2012 to support their development and raise their profile but only if 100 other public figures in London will join me in supporting other clubs*. PledgeBank (3. April and 25. June). Retrieved 28. February, 2007, from http://london.pledgebank.com/Sportclubpatrons.

Blake, R.R., & J.S. Mouton. (1964). *The managerial grid: Key orientations for achieving production through people*. Houston: Gulf Publishing.

Blank, V. (1999). *Schriftschutz: Schutz typographischer Schriftzeichen und Schriften im schweizerischen Immaterialgüter- und Lauterkeitsrecht*, from http://www.vblank.ch/recht/Schriftschutz.pdf.

Blau, P.M. (1963). *The dynamics of bureaucracy: A study of interpersonal relations in two government agencies* (rev. 2 ed.). Chicago: University of Chicago Press.

Blau, P.M. (1964). *Exchange and Power in Social Life*. New York: J. Wiley.

Blau, P.M. (1970). Formal theory of differentiation in organizations. *American Sociological Review, 35*(2), 201-218.

Blight, J.G., & J.M. Lang. (2005). *The fog of war: Lessons from the life of Robert S. McNamara*. Lanham: Rowman & Littlefield.

BLS. (2001). *Standard occupational classification: Computer and mathematical occupations - computer specialists* (16. October). Retrieved 5. March, 2007, from http://www.bls.gov/soc/soc_c0a0.htm.

BLS. (2006a). *Occupational outlook handbook: Computer programmers* (4. August). Retrieved 9. June, 2007, from http://www.bls.gov/oco/ocos110.htm.

BLS. (2006b). *Occupational outlook handbook: Computer software engineers* (4. August). Retrieved 9. June, 2007, from http://www.bls.gov/oco/ocos267.htm.

Blundell-Wignall, A. (2008). *The subprime crisis: Size, deleveraging and some policy options* (Financial market trends). Paris: OECD.

BMfSFJ. (2005). *Gender-Datenreport*. München: BMfSFJ.

Boehm, B.W. (1988). A spiral model of software development and enhancement. *Computer, 21*(5), 61-72.

Bollen, K.A., & K.F. Ting. (2000). A tetrad test for causal indicators. *Psychological Methods, 5*(1), 3-22.

Bollinger, T., E.S. Raymond, & W.C. Trader. (1999). Linux and open source success: Interview with Eric Raymond. *IEEE Software, 16*(1), 85-89.

Bolton, G.E., E. Katok, & R. Zwick. (1998). Dictator game giving: Rules of fairness versus acts of kindness. *International Journal of Game Theory, 27*(2), 269-299.

Bonaccorsi, A., & C. Rossi. (2003). Comparing motivations of individual programmers and firms to take part in the Open Source movement: From community to business. *SSRN Working Paper Series, 460861*, 1-33.

Bonaccorsi, A., & C. Rossi. (2004). Altruistic individuals, selfish firms? The structure of motivation in Open Source software. *First Monday, 9*(1).

Bonacich, P. (1976). Cooperation: An experimental analysis. *Contemporary Sociology, 5*(5), 689.

Bonanno, G. (2004). Memory and perfect recall in extensive games. *Games and Economic Behavior, 47*(2), 237-256.

Bond, M.H., K. Leung, A. Au, K.K. Tong, S.R. de Carrasquel, F. Murakami, et al. (2004). Culture-level dimensions of social axioms and their correlates across 41 cultures. *Journal of Cross-Cultural Psychology, 35*(5), 548-570.

Bond, R., & P.B. Smith. (1996). Culture and conformity: A meta-analysis of studies using Asch's (1952b, 1956) line judgment task. *Psychological Bulletin, 119*(1), 111-137.

Bong, M. (1996). Problems in academic motivation research and advantages and disadvantages of their solutions. *Contemporary Educational Psychology, 21*(2), 149-165.

Bortz, J. (1993). *Statistik für Sozialwissenschaftler* (4 ed.). Berlin: Springer.

Bortz, J., & N. Döring. (1995). *Forschungsmethoden und Evaluation* (2 ed.). Berlin: Springer.

Bosman, R., & F. van Winden. (2002). Emotional hazard in a power-to-take experiment. *Economic Journal, 112*(476), 147-169.

Bourne, S. (2004). A conversation with Bruce Lindsay *ACM Queue, 2*(8), 22-33.

Bowles, S., & H. Gintis. (2003). Prosocial emotions. *Santa Fe Institute Working Paper, 02-07-028*, 1-27.

Boyd, R., & P.J. Richerson. (1988). The evolution of reciprocity in sizable groups. *Journal of Theoretical Biology, 132*(3), 337-356.

Brandts, J., & A. Schram. (2001). Cooperation and noise in public goods experiments: applying the contribution function approach. *Journal of Public Economics, 79*(2), 399-427.

Breed, G.A. (2005, 2. September). Explosions fill New Orleans sky with smoke. *Associated Press*.

Bresnahan, T.F. (2001). Network effects and Microsoft. *Working Paper Stanford University*, 1-32.

Brewer, M. (2005). *Open source licensing: BSD is a better model* (07. April). Retrieved 12. September, 2006, from http://www.networkworld.com/supp/2005/opensource/070405-face-off-yes.html.

Brewer, M.B., & R.M. Kramer. (1986). Choice behavior in social dilemmas: Effects of social identity, group-size, and decision framing. *Journal of Personality and Social Psychology, 50*(3), 543-549.

Briscoe, B., A. Odlyzko, & B. Tilly. (2006). Metcalfe's law is wrong. *IEEE Spectrum, 43*(7), 34-39.

Broersen, P.M.T. (1986). Subset regression with stepwise directed search. *Applied Statistics: Journal of the Royal Statistical Society - Series C, 35*(2), 168-177.

Broersma, M. (2005). *Open source patent pool mooted and booted*. Techworld (12. August). Retrieved 12. February, 2007, from http://www.techworld.com/applications/news/index.cfm?NewsID=4209.

Brooks, F.P. (1987). No silver bullet: Essence and accidents of software engineering. *Computer, 20*(4), 10-19.

Brooks, F.P. (1995). *The mythical man-month: Essays on software engineering* (Anniversary ed.). Reading: Addison-Wesley.

Brown, J.H., T.G. Whitham, S.K.M. Ernest, & C.A. Gehring. (2001). Complex species interactions and the dynamics of ecological systems: Long-term experiments. *Science, 293*(5530), 643-650.

Brown, Z. (2003). Zack's kernel news. *Linux Magazine*(27), 16-17.

Bruner, G.C., P.J. Hensel, & K.E. James. (1992). *Marketing scales handbook: A compilation of multi-item measures* (Vol. 2). Chicago: American Marketing Association.

Brytting, T. (1986). The management of distance in antiquity. *Scandinavian Journal of Management Studies, 3*(2), 139-155.

BStU. (2007). *Der Volksaufstand vom 17. Juni 1953: Von den Ursachen bis zu den Spätfolgen*. BSTU. Retrieved 29. March, 2007, from http://www.bstu.bund.de/cln_043/nn_712860/DE/MfS-DDR-Geschichte/17-juni-1953/Ursachen-bis-Spaetfolgen/ursachen-bis-spaetfolgen_node.html_nnn=true.

Buchanan, J.M. (1967). *Public finance in democratic process; fiscal institutions and the individual choice*. Chapel Hill,: University of North Carolina Press.

Buchanan, J.M. (1971). *The bases for collective action*. New York: General Learning Press.

Buchanan, J.M. (1978). Markets, states, and extent of morals. *American Economic Review, 68*(2), 364-368.

Buckley, W., T. Burns, & L.D. Meeker. (1974). Structural resolutions of collective action problems. *Behavioral Science, 19*(5), 277-297.

Budescu, D.V., W.T. Au, & X.P. Chen. (1997). Effects of protocol of play and social orientation on behavior in sequential resource dilemmas. *Organizational Behavior and Human Decision Processes, 69*(3), 179-193.

Buhl, J., D.J.T. Sumpter, I.D. Couzin, J.J. Hale, E. Despland, E.R. Miller, et al. (2003). From disorder to order in marching locusts. *Science, 312*(5778), 1402-1406.

Bundesrepublik Deutschland. (2006). *Transfusionsgesetz* Bundesministerium der Justiz (31. October). Retrieved 30. January, 2007, from http://bundesrecht.juris.de/tfg/BJNR175200998.html.

Burgelman, R.A. (1983). A model of the interaction of strategic behavior, corporate context, and the concept of strategy. *Academy of Management Review, 8*(1), 61-70.

Burks, S.V., J.P. Carpenter, & E. Verhoogen. (2003). Playing both roles in the trust game. *Journal of Economic Behavior & Organization, 51*(2), 195-216.

Burnham, K.P., & D.R. Anderson. (2002). *Model selection and multimodel inference: A practical information-theoretic approach* (2 ed.). New York: Springer.

Burnham, K.P., & D.R. Anderson. (2004). Multimodel inference: Understanding AIC and BIC in model selection. *Sociological Methods and Research, 33*(2), 261-304.

Burt, R.S., & M. Knez. (1996). Trust and third party gossip. In: R. M. Kramer & T. R. Tyler (Eds.), *Trust in organizations: Frontiers of theory and research* (pp. 68-89). Thousand Oaks: Sage.

Butler, B.S. (2001). Membership size, communication activity, and sustainability: A resource-based model of online social structures. *Information Systems Research, 12*(4), 346-362.

Cabrera, A., & E.F. Cabrera. (2002). Knowledge-sharing dilemmas. *Organization Studies, 23*(5), 687-710.

Calabrese, E.J., & L.A. Baldwin. (2003). The hormetic dose-response model is more common than the threshold model in toxicology. *Toxicological Sciences, 71*(2), 246-250.

Camazine, S. (2001). *Self-organization in biological systems*. Princeton: Princeton University Press.

Camerer, C.F. (1997). Progress in behavioral game theory. *Journal of Economic Perspectives, 11*(4), 167-188.

Cameron, J., K.M. Banko, & W.D. Pierce. (2001). Pervasive negative effects of rewards on intrinsic motivation: The myth continues. *The Behavior Analyst, 24*(1), 1-44.

Cameron, J., & W.D. Pierce. (2002). *Rewards and intrinsic motivation: Resolving the controversy*. Westport: Bergin & Garvey.

Camilleri, S.F. (1962). Theory, probability, and induction in social research. *American Sociological Review, 27*(2), 170-178.

Caminer, D.T. (2003). Behind the curtain at LEO: A personal reminiscence. *IEEE Annals of the History of Computing, 25*(2), 3-13.

Campbell-Kelly, M. (2003). *From airline reservations to Sonic the Hedgehog: A history of the software industry*. Cambridge: MIT Press.

Cangelosi, V.E., & W.R. Dill. (1965). Organizational learning: Observations toward a theory. *Administrative Science Quarterly, 10*(2), 175-203.

Carlyon, D. (2001). *Dan Rice: The most famous man you've never heard of*. New York: Public Affairs.

Carnesale, A., & C. Glaser. (1982). ICBM vulnerability: The cures are worse than the disease. *International Security, 7*(1), 70-85.

Cartwright, D., & A.F. Zander (Eds.). (1960). *Group dynamics: Research and theory* (2 ed.). Evanston: Row Peterson & Company.

Castel, P., S. Negrier, J.P. Boissel, & P.d.A.à.l.R.C.e.C.d.l.r. Rhône-Alpes. (2006). Why don't cancer patients enter clinical trials? A review. *European Journal of Cancer, 42*(12), 1744-1748.

Castle, M., & R. Taras. (2002). *Democracy in Poland* (2 ed.). Boulder: Westview Press.

CEC. (2006). *Nuclear energy in California* (7. March). Retrieved 8. March, 2007, from http://www.energy.ca.gov/nuclear/overview.html.

Ceci, S.J., & E.L. Kain. (1982). Jumping on the bandwagon with the underdog: The impact of attitude polls on polling behavior. *Public Opinion Quarterly, 46*(2), 228-242.

Centers for Medicare & Medicaid Services. (2005). *An overview of the U.S. healthcare system: Facts and figures 1980-2000*. Baltimore: Centers for Medicare & Medicaid Services.

Chamberlin, W.H. (1935). *The Russian revolution 1917-1921* (Vol. 1). New York: MacMillan.

Chamley, C. (2004). *Rational herds: Economic models of social learning*. Cambridge: Cambridge University Press.

Chan, K.S., R. Godby, S. Mestelman, & R.A. Muller. (1997). Equity theory and the voluntary provision of public goods. *Journal of Economic Behavior & Organization, 32*(3), 349-364.

Chandler, A.D. (1962). *Strategy and structure: Chapters in the history of the industrial enterprise*. Cambridge: MIT Press.

Chao-Kuei, H. *Diagram of different software categories*. Free Software Foundation. Retrieved 15. June, 2006, from http://www.gnu.org/philosophy/categories.html.

Chao, A., & J.B. Schor. (1998). Empirical tests of status consumption: Evidence from women's cosmetics. *Journal of Economic Psychology, 19*(1), 107-131.

Chapman, M. (2006). *Britain warns US over jet software codes: £12bn Joint Strike Fighter order could be scrapped.* Vnunet.com (15. March). Retrieved 26. February, 2007, from http://www.vnunet.com/vnunet/news/2152035/joint-strike-fighter.

Charness, G., & M. Rabin. (2002). Understanding social preferences with simple tests. *Quarterly Journal of Economics, 117*(3), 817-869.

Charoencholvanich, K., & B. Pongcharoen. (2005). Oxford knee score and SF-36: Translation & reliability for use with total knee arthroscopy patients in Thailand. *Journal of The Medical Association of Thailand, 88*(9), 1194-1202.

Charter, R.A. (2003). A breakdown of reliability coefficients by test type and reliability method, and the clinical implications of low reliability. *Journal of General Psychology, 130*(3), 290-304.

Chen, M.K., & M. Hauser. (2005). Modeling reciprocation and cooperation in primates: evidence for a punishing strategy. *Journal of Theoretical Biology, 235*(1), 5-12.

Chen, M.K., V. Lakshminarayanan, & L.R. Santos. (2006). How basic are behavioral biases? Evidence from capuchin monkey trading behavior. *Journal of Political Economy, 114*(3), 517-537.

Chen, R.B., K.A. Wong, & H.C. Lee. (2001). Age, period, and cohort effects on life insurance purchases in the US. *Journal of Risk and Insurance, 68*(2), 303-327.

Chen, R.T. (1999). Vaccine risks: Real, perceived and unknown. *Vaccine, 17*(Supplement 3), 41-46.

Chen, Y.R., J. Brockner, & J. Greenberg. (2003). When is it "a pleasure to do business with you?" - The effects of relative status, outcome favorability, and procedural fairness. *Organizational Behavior and Human Decision Processes, 92*(1-2), 1-21.

Child, J., & A. Kieser. (1981). Development of organizations over time. In: P. C. Nystrom & W. H. Starbuck (Eds.), *Handbook of organizational design* (Vol. 1, pp. 28-64). Oxford: Oxford University Press.

Child, J., & S.B. Rodrigues. (2003). Corporate governance and new organizational forms: Issues of double and multiple agency. *Journal of Management and Governance, 7*(4), 337-360.

Chong, D. (1991). *Collective action and the civil rights movement.* Chicago: University of Chicago Press.

Christensen, C.M. (1997). *The innovator's dilemma: When new technologies cause great firms to fail.* Boston: Harvard Business School Press.

Christensen, C.M., & M.E. Raynor. (2003). *The innovater's solution: Creating and sustaining successful growth.* Boston: Harvard Business School Press.

Churchill, G.A. (1979). Paradigm for developing better measures of marketing constructs. *Journal of Marketing Research, 16*(1), 64-73.

CIA. (2007a). *The world factbook: Field listing - GDP per capita* (23. January). Retrieved 6. February, 2007, from https://www.cia.gov/cia/publications/factbook/fields/2004.html.

CIA. (2007b). *The world factbook: Field listing - Internet users* (23. January). Retrieved 6. February, 2007, from https://www.cia.gov/cia/publications/factbook/fields/2153.html.

CIA. (2007c). *The world factbook: Field listing - Languages* (8. January). Retrieved 5. March, 2007, from https://www.cia.gov/cia/publications/factbook/fields/2098.html.

CIA. (2007d). *The world factbook: Field listing - Population* (23. January). Retrieved 6. February, 2007, from https://www.cia.gov/cia/publications/factbook/fields/2119.html.

CIA. (2007e). *The world factbook: Field listing - Unemployment rate* (23. January). Retrieved 6. February, 2007, from https://www.cia.gov/cia/publications/factbook/fields/2129.html.

Cicchetti, D. (1994). Guidelines, criteria, and rules of thumb for evaluating normed and standardized assessment instruments in psychology. *Psychological Assessment, 6*(4), 284-290.

Cirka, C.C. (2005). Organizational citizenship behaviors as social dilemmas: Theoretical and methodological developments. In: D. L. Turnipseed (Ed.), *When actions speak as loudly as words: Autonomy support, psychological empowerment and organizational citizenship behavior* (pp. 291-327). New York: Nova Science Publishers.

Clary, E.G., & M. Snyder. (1999). The motivations to volunteer: Theoretical and practical considerations. *Current Directions in Psychological Science, 8*(5), 156-159.

Clary, E.G., M. Snyder, R.D. Ridge, J. Copeland, A.A. Stukas, J. Haugen, et al. (1998). Understanding and assessing the motivations of volunteers: A functional approach. *Journal of Personality and Social Psychology, 74*(6), 1516-1530.

Clinton, H. (2007). *Video transcript: Presidential exploratory committee announcement*. Hillary Clinton for President Exploratory Committee (20. January). Retrieved 24. January, 2007, from http://www.hillaryclinton.com/news/release/view/?id=1234.

Clucas, I. (1997). *A study of the options for utilization of bycatch and discards from marine capture fisheries*. FAO (FAO Fisheries Circular 928). Retrieved 27. November, 2006, from http://www.fao.org/docrep/w6602e/w6602E00.HTM.

CNN. (2007). *Frequently asked questions about I-Reports*. Retrieved 29. May, 2007, from http://edition.cnn.com/help/i-report.html.

Coase, R.H. (1960). The problem of social cost. *Journal of Law & Economics, 3*(Oct), 1-44.

Coase, R.H. (1974). Lighthouse in economics. *Journal of Law & Economics, 17*(2), 357-376.

Cochran, W.G. (1954). Some methods for strengthening the common X2 tests. *Biometrics, 10*(4), 417-451.

Coffee, J.C. (2001). Do norms matter? A cross-country evaluation. *University of Pennsylvania Law Review, 149*(6), 2151-2177.

Cohen, D.J. (1965). Some statistical issues in psychological research. In: B. B. Woleman (Ed.), *Handbook of Clinical Psychology* (pp. 95-121). New York: McGraw-Hill.

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2 ed.). Hillsdale: L. Erlbaum Associates.

Cohen, J. (1990). Things I have learned (so far). *American Psychologist, 45*(12), 1304-1312.

Cohen, J. (1992). A power primer. *Psychological Bulletin, 112*(1), 155-159.

Cohen, J. (2003). *Applied multiple regression/correlation analysis for the behavioral sciences* (3 ed.). Mahwah: Erlbaum.

Cohen, W.L., & D.M. Levinthal. (1990). Absorptive capacity: A new perspective on learning and innovation. *Administrative Science Quarterly, 35*, 128-152.

Colella, A. (2001). Coworker distributive fairness judgments of the workplace accommodation of employees with disabilities. *Academy of Management Review, 26*(1), 100-116.

Collins, H.M. (1993). The structure of knowledge. *Social Research, 60*(1), 95-116.

Comrey, A.L. (1988). Factor-analytic methods of scale development in personality and clinical-psychology. *Journal of Consulting and Clinical Psychology, 56*(5), 754-761.

Comte, A. (2000). *The catechism of positive religion*. Chestnut Hill: Adamant Media Corporation.

Condorcet, J.A.N.d.C. (1785). *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix*. Paris: De l'Imprimerie royale.

Connolly, T., L.M. Jessup, & J.S. Valacich. (1990). Effects of anonymity and evaluative tone on idea generation in computer mediated groups. *Management Science, 36*(6), 689-703.

Conrath, D.W. (1972). Sex role and cooperation in game of chicken. *Journal of Conflict Resolution, 16*(3), 433-443.

Conway, L.G., S.M. Sexton, & R.G. Tweed. (2006). Collectivism and governmentally initiated restrictions - A cross-sectional and longitudinal analysis across nations and within a nation. *Journal of Cross-Cultural Psychology, 37*(1), 20-41.

Conway, M.E. (1968). *How do committees invent? (Reprint from Datamation 1968, 14(4), 28-3)*. Retrieved 13. September, 2006, from http://www.melconway.com/research/committees.html.

Coombs, C.H. (1948). Some hypotheses for the analysis of qualitative variables. *Psychological Review, 55*(3), 167-174.

Coombs, C.H. (1966). Theory and methods of social measurements. In: L. Festinger & D. Katz (Eds.), *Research methods in the behavioral sciences* (pp. 471-535). New York: Holt, Rinehart & Winston.

Cordery, J.L., W.S. Mueller, & L.M. Smith. (1991). Attitudinal and behavioral effects of autonomous group working: A longitudinal field study. *Academy of Management Journal, 34*(2), 464-476.

Cordingly, D. (1997). *Under the black flag : the romance and the reality of life among the pirates*. Washington: Harvest Book.

Corneo, G., & O. Jeanne. (1997). Snobs, bandwagons, and the origin of social customs in consumer behavior. *Journal of Economic Behavior & Organization, 32*(3), 333-347.

Cornes, R., & T. Sandler. (1996). *The theory of externalities, public goods, and club goods* (2 ed.). Cambridge: Cambridge University Press.

Cosmides, L. (1989). The logic of social exchange: Has natural selection shaped how humans reason? Studies with the Wason selection task. *Cognition, 31*(3), 187-276.

Cosmides, L., J. Tooby, L. Fiddick, & G.A. Bryant. (2005). Detecting cheaters. *Trends in Cognitive Sciences, 9*(11), 505-506.

Covertex. (2003). *Soccer stadium "Allianz Arena", Munich, Germany: Project datasheets*. Retrieved 23. November, 2006, from http://www.covertex.de/version01/eng/downloads/.

Cowen, T., & D. Sutter. (1999). The costs of cooperation. *Review of Austrian Economics, 12*(2), 161-173.

Cox, G.W., & J.N. Katz. (2007). Gerrymandering roll calls in Congress, 1879-2000. *American Journal of Political Science, 51*(1), 108-119.

CPAN. (2006). *Comprehensive Perl archive network*. Retrieved 13. September, 2006, from http://www.cpan.org/index.html.

Crane, J. (1991). The epidemic theory of ghettos and neighborhood effects on dropping out and teenage childbearing. *American Journal of Sociology, 96*(5), 1226-1259.

Crawford, W.R. (1968). Item difficulty as related to complexity of intellectual processes. *Journal of Educational Measurement, 5*(2), 103-107.

Cronbach, L.J. (1951). Coefficient alpha and the internal structure of tests. *Psychometrika, 16*(3), 297-334.

Cronbach, L.J. (1972). *The dependability of behavioral measurements theory of generalizability for scores and profiles*. New York: John Wiley.

Cronbach, L.J. (1990). *Essentials of psychological testing* (5 ed.). New York: Harper Collins.

Cronbach, L.J. (2004). My current thoughts on coefficient alpha and successor procedures, *CSE Report 643* (pp. 1-32). Los Angeles: University of California.

Cronbach, L.J., & P.E. Meehl. (1955). Construct validity in psychological tests. *Psychological Bulletin, 52*(4), 281-302.

Cropanzano, R., & Z.S. Byrne. (2000). Workplace justice and the dilemma of organizational citizenship. In: M. Van Vugt (Ed.), *Cooperation in modern society : promoting the welfare of communities, states, and organizations* (Vol. 1, pp. 142-161). London: Routledge.

Cropanzano, R., Z.S. Byrne, D.R. Bobocel, & D.E. Rupp. (2001). Moral virtues, fairness heuristics, social entities, and other denizens of organizational justice. *Journal of Vocational Behavior, 58*(2), 164-209.

Crowston, K., J. Howison, & H. Annabi. (2006). Information systems success in free and open source software development: Theory and measures. *Software Process: Improvement and Practice, 11*(2), 123-148.

Csikszentmihalyi, M. (1991). *Flow: The psychology of optimal experience*. New York: Harper Perennial.

Csikszentmihalyi, M. (2000). *Beyond boredom and anxiety* (25th anniversary ed.). San Francisco: Jossey-Bass Publishers.

Csikszentmihalyi, M., & J. Geirland. (1996). *Go with the flow*. Retrieved 25. October, 2006, from http://www.wired.com/wired/archive/4.09/czik.html.

Curtis, B. (1981). Substantiating programmer variability. *Proceedings of the IEEE, 69*(7), 846.

Curtis, B., H. Krasner, & N. Iscoe. (1988). A field study of the software design process for large systems. *Communications of the ACM, 31*(11), 1268-1287.

Cusumano, M. (2005). Foreword. In: J. Feller, B. Fitzgerald, S. Hissam & K. R. Lakhani (Eds.), *Perspectives on free and open source software* (pp. xi-xiii). Cambridge: MIT Press.

Dahlander, L., & M.G. Magnusson. (2005). Relationships between open source software companies and communities: Observations from Nordic firms. *Research Policy, 34*(4), 481-493.

Daniels, D., J. Joireman, & D. Kamdar. (2005). Organizational citizenship behaviors as social dilemmas: Theoretical and methodological developments. In: D. L. Turnipseed (Ed.), *Handbook of organizational citizenship behavior a review of "good soldier" activity in organizations* (pp. 79-106). New York: Nova Science Publishers.

David, P.A. (1985). Clio and the economics of QWERTY. *American Economic Review, 75*(2), 332-337.

David, P.A., A. Waterman, & S. Arora. (2003a). *FLOSS-US: Responses to the questionnaire*. Retrieved 6. September, 2006, from http://www.stanford.edu/group/floss-us/stats/.

David, P.A., A. Waterman, & S. Arora. (2003b). *FLOSS-US: The Free/libre/open source software survey for 2003*. Retrieved 6. September, 2006, from http://www.stanford.edu/group/floss-us/report/FLOSS-US-Report.pdf.

Davies, R.J., & O. Ikeno. (2002). *The Japanese mind: Understanding contemporary Japanese culture*. Boston: Tuttle Publishing.

Dawes, C.T., J.H. Fowler, T. Johnson, R. McElreath, & O. Smirnov. (2007). Egalitarian motives in humans. *Nature, 446*(7137), 794-796.

Dawes, R.M. (1980). Social dilemmas. *Annual Review of Psychology, 31*, 169-193.

Dawes, R.M., J. Mctavish, & H. Shaklee. (1977). Behavior, communication, and assumptions about other peoples behavior in a commons dilemma situation. *Journal of Personality and Social Psychology, 35*(1), 1-11.

Dawes, R.M., & D.M. Messick. (2000). Social dilemmas. *International Journal of Psychology, 35*(2), 111-116.

Dawes, R.M., & R.H. Thaler. (1988). Anomalies - Cooperation. *Journal of Economic Perspectives, 2*(3), 187-197.

Dawson, T. (2008). *Fermi Linux timeline*. Fermi Linux (18. July). Retrieved 24. August, 2008, from https://fermilinux.fnal.gov/documentation/about/brief-history.

Day, K.M., & R.A. Devlin. (1996). Volunteerism and crowding out: Canadian econometric evidence. *Canadian Journal of Economics, 29*(1), 37-53.

De Alessi, L. (1975). Toward an analysis of postdisaster cooperation. *American Economic Review, 65*(1), 127-138.

De Charms, R. (1968). *Personal causation: The internal affective determinants of behavior*. New York: Academic Press.

De Cremer, D., T.R. Tyler, & N. den Ouden. (2005). Managing cooperation via procedural fairness: The mediating influence of self-other merging. *Journal of Economic Psychology, 26*(3), 393-406.

De Dreu, C.K.W., L.R. Weingart, & S. Kwon. (2000). Influence of social motives on integrative negotiation: A meta-analytic review and test of two theories. *Journal of Personality and Social Psychology, 78*(5), 889-905.

De Goyeneche, J.-M., & E.A.F. De Sousa. (1999). Loadable kernel modules. *IEEE Software, 16*(1), 65-71.

De Mente, B.L. (2003). *Etiquette guide to Japan: Know the rules ... that make the difference* (10 ed.). Tokyo: Tuttle Publishing.

De Quervain, D.J.F., U. Fischbacher, V. Treyer, M. Schelthammer, U. Schnyder, A. Buck, et al. (2004). The neural basis of altruistic punishment. *Science, 305*(5688), 1254-1258.

De Souza, C., J. Froehlich, & P. Dourish. (2005). *Seeking the source: Software source code as a social and technical artifact.* Paper presented at the International ACM SIGGROUP Conference on Supporting Group Work, Sanibel Island.

De Vries, S., & H.A.M. Wilke. (1992). Constrained egoism and resource management under uncertainty. In: W. B. G. Liebrand, D. M. Messick & H. A. M. Wilke (Eds.), *Social dilemmas: Theoretical issues and research findings* (pp. 81-99). Oxford: Pergamon Press.

De Waal, F.B.M. (2007). With a little help from a friend. *PLOS Biology, 5*(7), 1406-1408.

Debian Project. (2004). *Debian social contract* (Version 1.1). Retrieved 3. January, 2007, from http://www.debian.org/social_contract.

Debian Project. (2007). *Debian bug report logs - #354622: Uses Mozilla Firefox trademark without permission* (12. February). Retrieved 13. February, 2007, from http://bugs.debian.org/cgi-bin/bugreport.cgi?bug=354622.

Deci, E.L. (1971). Effects of externally mediated rewards on intrinsic motivation. *Journal of Personality and Social Psychology, 22*, 113-120.

Deci, E.L., & W.F. Cascio. (1972). *Changes in intrinsic motivation as a function of negative feedback and threats.* Paper presented at the Eastern Psychological Association, Boston.

Deci, E.L., J.P. Connell, & R.M. Ryan. (1989). Self-determination in a work organization. *Journal of Applied Psychology, 74*(4), 580-590.

Deci, E.L., R. Koestner, & R.M. Ryan. (1999). A meta analytic review of experiments examining the effects of extrinsic rewards on intrinsic motivation. *Psychological Bulletin, 125*(6), 627-668.

Deci, E.L., & R.M. Ryan. (1985). *Intrinsic motivation and self-determination in human behavior*. New York: Plenum.

Deci, E.L., & R.M. Ryan. (2000). The "what" and "why" of goal pursuits: Human needs and the self-determination of behavior. *Psychological Inquiry, 11*(4), 227-268.

Deci, E.L., & R.M. Ryan. (2006). *Intrinsic motivation inventory* (27. October). Retrieved 30. April, 2007, from http://www.psych.rochester.edu/SDT/measures/intrins_scl.html.

Deci, E.L., R.M. Ryan, K.W. Brown, & A. Capobianco. (2006). *Basic psychological needs scales: Basic need satisfaction at work*. Retrieved 11 August, 2006, from http://www.psych.rochester.edu/SDT/measures/needs_scl.html.

Deci, E.L., R.M. Ryan, M. Gagne, D.R. Leone, J. Usunov, & B.P. Kornazheva. (2001). Need satisfaction, motivation, and well-being in the work organizations of a former eastern bloc country: A cross-cultural study of self-determination. *Personality and Social Psychology Bulletin, 27*(8), 930-942.

Defoe, D. (1719). *The life and strange surprizing adventures of Robinson Crusoe, of York, mariner: who lived eight and twenty years all alone in an un-inhabited island on the coast of America, near the mouth of the great river of Oroonoque; having been cast on shore by shipwreck, wherein all the men perished but himself. With and account how he was at last as strangely deliver'd by pyrates* (4 ed.). London: Printed for W. Taylor.

Defoe, D. (1724). *A general history of the pyrates* (2 ed.). London: Printed for, and sold by T. Warner.

Dellas, H., & A. Stockman. (1993). Self-fulfilling expectations, speculative attack, and capital controls. *Journal of Money Credit and Banking, 25*(4), 721-730.

Delucchi, K.L. (1983). The use and misuse of chi-square: Lewis and Burke revisited. *Psychological Bulletin, 94*(1), 166-176.

Deming, W.E. (1957). *Some theory of sampling*. New York: Wiley.

Deming, W.E. (1960). *Sample design in business research*. New York: Wiley.

DeNardo, J. (1985). *Power in numbers : the political strategy of protest and rebellion*. Princeton: Princeton University Press.

Dening, G. (1992). *Mr. Bligh's bad language: Passion, power, and theatre on the Bounty*. Cambridge: Cambridge University Press.

DeTienne, K.B. (1994). Big brother is watching: Computer monitoring and communication. *IEEE Transactions on Professional Communication, 37*(1), 5-10.

Deutsch, M. (1949). A theory of co-operation and competition. *Human Relations, 2*(2), 129-152.

Deutsch, M. (1973). *The resolution of conflict; constructive and destructive processes*. New Haven: Yale University Press.

Deutsch, M. (1975). Equity, equality, and need: What determines which value will be used as the basis of distributive justice? *Journal of Social Issues, 31*(3), 137-149.

DeVellis, R.F. (2003). *Scale development: Theory and applications* (2 ed.). Thousand Oaks: Sage.

Diamantopoulos, A., & J.A. Siguaw. (2006). Formative versus reflective indicators in organizational measure development: A comparison and empirical illustration. *British Journal of Management, 17*(4), 263-282.

Diamantopoulos, A., & H.M. Winklhofer. (2001). Index construction with formative indicators: An alternative to scale development. *Journal of Marketing Research, 38*(2), 269-277.

Diedrich, T. (1991). *Der 17. Juni 1953 in der DDR: Bewaffnete Gewalt gegen das Volk*. Berlin: Dietz.

Dijkstra, E.W. (1972). The humble programmer. *Communications of the ACM, 15*(10), 859-866.

Ditlea, S. (2002). *Rewriting the Bible in 0's and 1's*. MIT Technology Review (11. January). Retrieved 05. January, 2006, from http://www.technologyreview.com/printer_friendly_article.aspx?id=11960.

DJI. (1989a). *Meinungsumfrage November 1989 - Meinungsumfrage November 1989 Codebuch*. GESIS (ZA). Retrieved 2. April, 2007, from http://www.za.uni-koeln.de/data/ddr-nbl/codebuch/6018cb.pdf.

DJI. (1989b). *Montagsdemonstration in Leipzig am 4.12.1989: Codebuch*. GESIS (ZA). Retrieved 2. April, 2007, from http://www.za.uni-koeln.de/data/ddr-nbl/codebuch/6019cb.pdf.

Djuren, H. (1963). *Das Deichrecht im Lande Wursten: Seit Einführung der Generalkommunion (1661)*. Hildesheim: Lax.

DOJ. (1999). *United States v. Microsoft (Civil Action No. 98-1232 and 98-1233): Plaintiff's joint proposed findings of fact*. U.S. Department of Justice. Retrieved 21. November, 2006, from http://www.usdoj.gov/atr/cases/f2600/2613.htm.

Domen, R.E. (1995). Paid-versus-volunteer blood donation in the United States: A historical review. *Transfusion Medicine Reviews, 9*(1), 53-59.

Donovan, J.J. (2001). Work Motivation. In: N. Anderson, D. S. Ones, H. K. Sinangil & C. Viswesvaran (Eds.), *Handbook of industrial, work and organizational psychology* (Vol. 2, pp. 53-76). London: Sage.

Dore, R. (2005). Deviant or different? Corporate governance in Japan and Germany. *Corporate Governance: An International Review, 13*(3), 437-446.

Dougherty, D., D. Sims, & M. Tiemann. (2000). *Will money spoil open source?* O'Reilly Network. Retrieved 20. August, 2002, from http://linux.oreillynet.com/pub/a/linux/2000/01/31/interview/.

Douglas, T. (2006). *How 7/7 'democratised' the media*. BBC News. Retrieved 29. May, 2007, from http://news.bbc.co.uk/2/hi/uk_news/5142702.stm.

Douthwaite, B., J.D.H. Keatinge, & J.R. Park. (2001). Why promising technologies fail: The neglected role of user innovation during adoption. *Research Policy, 30*(5), 819-836.

Downs, A. (1957). *An economic theory of democracy*. New York: Harper & Brothers.

Downs, A. (1967). The life cycle of bureaus. In: A. Downs (Ed.), *Inside bureaucracy* (pp. 269-309). San Francisco: Little, Brown & Company.

Dresher, M. (1961). *Games of strategy: Theory and applications*. Englewood Cliffs: Prentice-Hall.

Driessen, P.A. (1985). The crowding out effect of governmental transfers on private charitable contributions: Comment. *National Tax Journal, 38*(4), 571-573.

Drucker, P. (2001). The next society. *The Economist, 361*(8246), 3-5.

Dubner, S.J., & S.D. Levitt. (2005). Monkey business: Can capuchins understand money? *The New York Times Magazine, 5. June*, 30-31.

Dudycha, A.L., & J.B. Carpente. (1973). Effects of item format on item discrimination and difficulty. *Journal of Applied Psychology, 58*(1), 116-121.

Dunbar, R.L.M. (1981). Designs for organizational control. In: P. C. Nystrom & W. H. Starbuck (Eds.), *Handbook of organizational design* (Vol. 2, pp. 85-115). Oxford: Oxford University Press.

Dunc-Tank. (2006, 19. September). *What does "Dunc" mean?* Retrieved 3. January, 2007, from http://www.dunc-tank.org/about.html.

Durkheim, â. (1902). *De la division fu travail social* (2 ed.). Paris: F. Alcan.

Dwyer, G.P., & C.M. Lindsay. (1984). Giffen, Robert and the Irish potato. *American Economic Review, 74*(1), 188-192.

Dyba, T., V.B. Kampenes, & D.I.K. Sjoberg. (2006). A systematic review of statistical power in software engineering experiments. *Information and Software Technology, 48*(8), 745-755.

Dyer, L., & D.F. Parker. (1975). Classifying outcomes in work motivation research: Examination of intrinsic-extrinsic dichotomy. *Journal of Applied Psychology, 60*(4), 455-458.

Eastlund, T. (1998). Monetary blood donation incentives and the risk of transfusion-transmitted infection. *Transfusion, 38*(9), 874-882.

Eastlund, T. (2003). Willingness of volunteer blood donors to be volunteer semen donors. *Fertility and Sterility, 80*(6), 1513-1514.

Eastman, K.K., & B.S. Pawar. (2005). An integrative view of and a common conceptual space for employee extra-role behaviors. In: D. L. Turnipseed (Ed.), *Handbook of organizational citizenship behavior a review of "good soldier" activity in organizations* (pp. 25-46). New York: Nova Science Publishers.

Ebel, R.L. (1967). Relation of item discrimination to test reliability. *Journal of Educational Measurement, 4*(3), 125-128.

ECB. (2006). *Euro foreign exchange reference rates: Previous rates*. Retrieved 29. August, 2006, from http://www.ecb.int/stats/exchange/eurofxref/html/index.en.html.

Economist. (1998). The perils of percentages. *Economist, 346*(8064), 70.

Economist. (2000). Deadly embrace. *The Economist, 355*(8164), 58-59.

Economist. (2007). Down the Matterhorn *Economist, 384*(8537), 77-78.

Eddington, A. (1941). *The philosophy of physical science*. New York: Macmillan.

Edison, T.A. (1892). Process of duplicating phonograms. USPTO 484,582: Edison Phonograph Company.

Edwards, A.L., & F.P. Kilpatrick. (1948). A technique for the construction of attitude scales. *Journal of Applied Psychology, 32*(4), 374-384.

Edwards, J.R., & R.P. Bagozzi. (2000). On the nature and direction of relationships between constructs and measures. *Psychological Methods, 5*(2), 155-174.

Eek, D., & A. Biel. (2003). The interplay between greed, efficiency, and fairness in public goods dilemmas. *Social Justice Research, 16*(3), 195-215.

Eek, D., A. Biel, & T. Garling. (1999). The fairness effect on cooperation in asymmetric social dilemmas when equality is perceived as unfair. *Göteborg Psychological Reports, 29*(2), 1-21.

Egyedi, T.M., & R.v. Wendel de Joode. (2003). *Standards and coordination in open source software.* Paper presented at the 3rd IEEE Conference on Standardization and Innovation in Information Technology, Delft.

Eichberger, J. (1993). *Game theory for economists*. San Diego: Academic Press.

Eichenberger, R., & B.S. Frey. (1990). Entscheidungsanomalien. *Wirtschaftswissenschaftliches Studium, 19*(6), 270-274.

Einstein, A. (1949). Autobiographical notes - Autobiographisches. In: P. A. Schilpp (Ed.), *Albert Einstein: Philosopher - Scientist* (pp. 1-95). Evanston: Library of Living Philosophers.

Elenkov, D.S. (1998). Can American management concepts work in Russia? A cross-cultural comparative study. *California Management Review, 40*(4), 133-156.

Ellickson, R.C. (1993). Property in land. *Yale Law Journal, 102*(6), 1315-1400.

Ellsberg, D. (1961). Risk, ambiguity, and the Savage axioms. *Quarterly Journal of Economics, 75*(4), 643-669.

Ellsberg, D. (1963). Risk, ambiguity, and the Savage axioms: Reply. *Quarterly Journal of Economics, 77*(2), 336-342.

Elm, K., & P. Feige. (1980). Der Verfall des zisterziensischen Ordensleben im späten Mittelalter. In: K. Elm, P. Joerissen & H. J. Roth (Eds.), *Die Zisterzienser: Ordensleben zwischen Ideal und Wirklichkeit* (Vol. 10, pp. 237-242). Bonn: Rheinland-Verlag.

Elster, J. (1989a). *The cement of society*. Cambridge: Cambridge University Press.

Elster, J. (1989b). *Nuts and bolts for the social sciences*. Cambridge: Cambridge University Press.

Elster, J. (1998). Emotions and economic theory. *Journal of Economic Literature, 36*(1), 47-74.

Emlen, S.T. (1982). The evolution of helping: I. An ecological constraints model. *American Naturalist, 119*(1), 29-39.

Emlen, S.T. (1991). Evolution of cooperative breeding in birds and mammals. In: J. R. Krebs & N. B. Davies (Eds.), *Behavioural ecology an evolutionary approach* (3 ed., pp. 301-337). Oxford: Blackwell.

Engelfriet, C.P., H.W. Reesink, A. Farrugia, D. Schonitzer, S. Wendel, S. Olyntho, et al. (2006). Paid vs. unpaid donors. *Vox Sanguinis, 90*(1), 63-70.

Engelhart, M.D. (1965). A comparison of several item discrimination indices. *Journal of Educational Measurement, 2*(1), 69-76.

Erev, I., G. Bornstein, & R. Galili. (1993). Constructive intergroup competition as a solution to the free rider problem: A field experiment. *Journal of Experimental Social Psychology, 29*(6), 463-478.

Erev, I., & A. Rapoport. (1990). Provision of step level public goods: The sequential contribution mechanism. *Journal of Conflict Resolution, 34*(3), 401-425.

Erlei, M. (2003). Heterogeneous social preferences. *Clausthal University of Technology Working Paper*, 1-29.

Eurostat, & T. Meri. (2007). Doctorate holders. *Statistics in focus: Science and technology, 131*, 1-7.

Faber, B.D. (2002). *Educational models and open source: resisting the proprietary university.* Paper presented at the International Conference on Computer Documentation, Toronto.

Falk, A., & M. Kosfeld. (2006). The hidden costs of control. *American Economic Review, 96*(5), 1611-1630.

FAO. (2006). *Locust and other migratory pests group: FAQ*. FAO. Retrieved 7. June, 2006, from http://www.fao.org/ag/locusts/en/info/info/faq/index.html.

Farrell, J., & P. Klemperer. (2006). Coordination and lock-In: Competition with switching costs and network effects *Competition Policy Center Working Paper, CPC06-058*.

Fayers, P.M., D.J. Hand, K. Bjordal, & M. Groenvold. (1997). Causal indicators in quality of life research. *Quality of Life Research, 6*(5), 393-406.

FDA. (2001). *FDA approves first nucleic acid test (NAT) systems to screen plasma for human immunodeficiency virus (HIV) and hepatitis C virus (HCV)*. FDA Talk Paper T01-42 (21. September). Retrieved 29. January, 2007, from http://www.fda.gov/bbs/topics/ANSWERS/2001/ANS01103.html.

FDA. (2006). *Compliance policy guides manual: Chapter 2: Biologics: Sub chapter 230.150: Blood donor classification statement, paid or volunteer donor*. Office of Regulatory Affairs (30. November). Retrieved 29. January, 2007, from http://www.fda.gov/ora/compliance_ref/cpg/cpgbio/cpg230-150final.htm.

Federal Election Commission. (2005, May). *Federal elections 2004*, from http://www.fec.gov/pubrec/fe2004/federalelections2004.pdf.

Fehr, E., & U. Fischbacher. (2004). Social norms and human cooperation. *Trends in Cognitive Sciences, 8*(4), 185-190.

Fehr, E., & H. Gintis. (2007). Human motivation and social cooperation: Experimental and analytical foundations. *Annual Review of Sociology, 33*, 43-64.

Fehr, E., & K. Schmidt. (1999). A theory of fairness, competition, and cooperation. *Quarterly Journal of Economics, 114*(3), 817-868.

Feller, J., & B. Fitzgerald. (2002). *Understanding open source software development*. London: Addison-Wesley.

Feller, J., B. Fitzgerald, S. Hissam, & K.R. Lakhani. (2005). Introduction. In: J. Feller, B. Fitzgerald, S. Hissam & K. R. Lakhani (Eds.), *Perspectives on free and open source software* (pp. xvii-xxxi). Cambridge: MIT Press.

Ferguson, E.D., & S. Schmitt. (1988). Gender linked stereotypes and motivation affect performance in the prisoners dilemma game. *Perceptual and Motor Skills, 66*(3), 703-714.

Ferguson, T.D., & D.J. Ketchen. (1999). Organizational configurations and performance: The role of statistical power in extant research. *Strategic Management Journal, 20*(4), 385-395.

Feynman, R.P. (1982). Simulating physics with computers. *International Journal of Theoretical Physics, 21*(6-7), 467-488.

Feynman, R.P. (1986). Appendix F: Personal observations on the reliability of the shuttle. In: Presidential Commission on the Space Shuttle Challenger Accident (Ed.), *Report to the President* (Vol. 2). Washington: The Commission.

Feynman, R.P., & R. Leighton. (1997). *"Surely you're joking, Mr. Feynman!": Adventures of a curious character*. New York: W.W. Norton & Company.

FFII. (2007a). *Foundation for a free information infrastructure*. Retrieved 13. February, 2007, from http://www.ffii.org.

FFII. (2007b). *Patented European webshop*. Retrieved 13. February, 2007, from http://webshop.ffii.org/.

Fielding, R.T. (1999). Shared leadership in the Apache project. *Communications of the ACM, 42*(4), 42-43.

Finch, J. (1987). The vignette technique in survey-research. *Sociology, 21*(1), 105-114.

Fischbacher, U., & S. Gächter. (2006). Heterogeneous social preferences and the dynamics of free riding in public goods. *IZA Discussion Paper Series, 2011*, 1-42.

Fischbacher, U., S. Gachter, & E. Fehr. (2001). Are people conditionally cooperative? Evidence from a public goods experiment. *Economics Letters, 71*(3), 397-404.

Fischer, R., M.C. Ferrieira, E.M.L. Assmar, P. Redford, & C. Harb. (2005). Organizational behaviour across cultures. *International Journal of Cross Cultural Management, 5*(1), 27-48.

Fisher, R., & W.P. Smith. (1969). Conflict of interest and attraction in development of cooperation. *Psychonomic Science, 14*(4), 154-&.

Fisher, R.A. (1918). The correlation between relatives on the supposition of Mendelian inheritance. *Philosophical Transactions of the Royal Society of Edinburgh 52*, 399-433.

Fisher, R.A. (1925). *Statistical methods for research workers*. Edinburgh: Oliver & Boyd.

Fisher, R.A. (1971). *The design of experiments* (8 ed.). New York: Hafner.

Fisher, R.J. (1993). Social desirability bias and the validity of indirect questioning. *Journal of Consumer Research, 20*(2), 303-315.

Fisher, R.J., & J.E. Katz. (2000). Social desirability bias and the validity of self-reported values. *Psychology & Marketing, 17*(2), 105-120.

Fitzgerald, B. (2005). Has Open Source Software a Future. In: J. Feller, B. Fitzgerald, S. Hissam & K. R. Lakhani (Eds.), *Perspectives on free and open source software* (pp. 93-106). Cambridge: MIT Press.

Fitzgerald, B., & J. Feller. (2001). Open source software: investigating the software engineering, psychosocial and economic issues. *Information Systems Journal, 11*(4), 273-276.

Fleitas, D.W. (1971). Bandwagon and underdog effects in minimal-information elections. *American Political Science Review, 65*(2), 434-&.

Fletcher, C., & E.L. Perry. (2001). Performance appraisal and feedback: A consideration of national culture and a review of contemporary research and future trends. In: N. Anderson, D. S. Ones, H. K. Sinangil & C. Viswesvaran (Eds.), *Handbook of industrial, work and organizational psychology* (Vol. 1, pp. 127-144). London: Sage.

Flohr, M. (2008). *DDR: Ein Sozialparadies, keine Diktatur*. Spiegel (25. July). Retrieved 26. July, 2008, from http://www.spiegel.de/schulspiegel/wissen/0,1518,567907,00.html.

Flood, M.M. (1952). Some experimental games. *Rand Research Memorandum, 789*.

Flood, M.M. (1958). Some experimental games. *Management Science, 5*(1), 5-26.

Folger, R. (1977). Distributive and procedural justice: Combined impact of voice and improvement on experienced inequity. *Journal of Personality and Social Psychology, 35*(2), 108-119.

Folger, R., & D.P. Skarlicki. (1998). A popcorn metaphor for workplace violence. In: R. W. Griffin, A. O'Leary-Kelly & J. M. Collins (Eds.), *Dysfunctional behavior in organizations: Violent and deviant behavior* (pp. 43-82). Stamford, Conn. ; London: JAI.

Fombrun, C.J. (1984). Structures of organizational governance. *Human Relations, 37*(3), 207-223.

Forden, G., P. Podvig, & T.A. Postol. (2000). False alarm, nuclear danger. *IEEE Spectrum, 37*(3), 31-39.

Forgas, J.P. (1998). On feeling good and getting your way: Mood effects on negotiator cognition and bargaining strategies. *Journal of Personality and Social Psychology, 74*(3), 565-577.

Fox, J., & M. Guyer. (1977). Group size and others strategy in an N-person Game. *Journal of Conflict Resolution, 21*(2), 323-338.

Franck, E., & C. Jungwirth. (2003a). Die Governance von Open Source Projekten. *Zeitschrift für Betriebswirtschaft, 73*(5), 1-21.

Franck, E., & C. Jungwirth. (2003b). Reconciling rent-seekers and donators: The governance structure of open source. *Journal of Management and Governance, 7*(4), 401-421.

Frank, R.H. (1993). The strategic role of the emotions: Reconciling oversocialized and undersocialized accounts of behavior. *Rationality and Society, 5*(2), 160-184.

Franke, N., & S. Shah. (2003). How communities support innovative activities: an exploration of assistance and sharing among end-users. *Research Policy, 32*(1), 157-178.

Franke, N., & E. von Hippel. (2003). Satisfying heterogeneous user needs via innovation toolkits: the case of Apache security software. *Research Policy, 32*(7), 1199-1215.

Franks, N.R., A.B. Sendova-Franks, & C. Anderson. (2001). Division of labour within teams of New World and Old World army ants. *Animal Behaviour, 62*(4), 635-642.

FreeBSD. (2005). *Committer's guide: 10 - The FreeBSD committers' big list of rules*. Retrieved 25. September, 2006, from http://www.freebsd.org/doc/en_US.ISO8859-1/articles/committers-guide/rules.html.

Freiberger, P., & M. Swaine. (1984). *Fire in the valley: The making of the personal computer*. Berkeley: Osborne/McGraw-Hill.

Freitag, M. (1996). Wahlbeteiligung in westlichen Demokratien. *Swiss Political Science Review, 2*(4), 101-134.

French, J.R.P., & B.H. Raven. (1960). The bases of social power. In: D. Cartwright & A. F. Zander (Eds.), *Group dynamics: Research and theory* (2 ed., pp. 607-623). Evanston: Row Peterson & Company.

Frese, E. (1995). *Grundlagen der Organisation: Konzept, Prinzipien, Strukturen* (6 ed.). Wiesbaden: Gabler.

Freshmeat. (2005). *Available sub-categories: License* (3. February). Retrieved 3. February, 2005, from http://freshmeat.net/browse/13/.

Freshmeat. (2006). *Available sub-categories: Development status* (11. July). Retrieved 11. July, 2006, from http://freshmeat.net/browse/6/.

Frey, B.S., & R. Jegen. (2001). Motivation crowding theory: A survey of empirical evidence. *Journal of Economic Surveys, 15*(5), 589-611.

Frey, B.S., & S. Meier. (2005). Selfish or indoctrinated economists? *European Journal of Law and Economics, 19*(2), 165-171.

Frey, B.S., & M. Osterloh. (2002). Motivation: A dual-edged factor of production. In: B. S. Frey & M. Osterloh (Eds.), *Successful management by motivation: balancing intrinsic and extrinsic incentives* (pp. 3-26). Berlin: Springer.

Frey, B.S., & B. Torgler. (2007). Tax morale and conditional cooperation. *Journal of Comparative Economics, 35*(1), 136-159.

Friedman, B.M. (1978). Crowding out or crowding in? Economic consequences of financing government deficits. *Brookings Papers on Economic Activity*(3), 593-654.

Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association, 32*(200), 675-701.

Friedman, M. (1966). *Essays in positive economics*. Chicago: University of Chicago Press.

Fritz, W.M. (1996). The women of ENIAC. *IEEE Annals of the History of Computing, 18*(3), 13-28.

Frohlich, N., J.A. Oppenheimer, & O.R. Young. (1971). *Political leadership and collective goods*. Princeton: Princeton University Press.

FSF. (2004, 20. May). *Licenses*. Free Software Foundation. Retrieved 25. May, 2004, from http://www.gnu.org/licenses/licenses.html.

FSF. (2005). *GNU Lesser General Public License*. Free Software Foundation. Retrieved 08. February, 2007, from http://www.gnu.org/copyleft/lesser.html.

FSF. (2006a). *The free software definition*. Free Software Foundation. Retrieved 13. June, 2006, from http://www.fsf.org/licensing/essays/free-sw.html.

FSF. (2006b). *FSF's opinion of the Apple Public Source License (APSL) 2.0*. Free Software Foundation. Retrieved 13. June, 2006, from http://www.gnu.org/philosophy/apsl.html.

FSF. (2006c). *GNU General Public License*. Free Software Foundation. Retrieved 13. June, 2006, from http://www.gnu.org/copyleft/gpl.html.

FSF. (2006d). *Selling free software*. Free Software Foundation. Retrieved 06. November, 2006, from http://www.gnu.org/philosophy/selling.html.

FSF. (2006e). *Various licenses and comments about them*. Free Software Foundation. Retrieved 13. June, 2006, from http://www.gnu.org/licenses/license-list.html.

FSF. (2006f). *What is copyleft?* Free Software Foundation. Retrieved 13. June, 2006, from http://www.gnu.org/copyleft/copyleft.html.

FSF. (2006g). *Why "free software" is better than "open source"*. Free Software Foundation. Retrieved 13. June, 2006, from http://www.gnu.org/philosophy/free-software-for-freedom.html.

FSF Europe. (2005). *Press release: Early comment on new Microsoft Shared Source Licenses*. Free Software Foundation. Retrieved 2006, 19. October, from http://mailman.fsfeurope.org/pipermail/press-release/2005q4/000120.html.

FSF Europe. (2006). *Software patents in Europe*. Free Software Foundation (28. November). Retrieved 2007, 12. February, from http://www.fsfeurope.org/projects/swpat/swpat.en.html.

FTC. (2003). *To promote innovation: The proper balance of competition and patent law and policy*. Washington: Federal Trade Commission.

Fudenberg, D., & J. Tirole. (1989). Noncooperative game theory for industrial organization: An introduction and overview. In: R. Schmalensee & R. D. Willig (Eds.), *Handbook of industrial organization* (Vol. 1, pp. 259-327). Amsterdam: North-Holland.

Fudenberg, D., & J. Tirole. (1991). *Game theory*. Cambridge: MIT Press.

Fulk, J., R. Heino, A.J. Flanagin, P.R. Monge, & F. Bar. (2004). A test of the individual action model for organizational information commons. *Organization Science, 15*(5), 569-585.

Fulk, T., A.J. Flanagin, M.E. Kalman, P.R. Monge, & T. Ryan. (1996). Connective and communal public goods in interactive communication systems. *Communication Theory, 6*(1), 60-87.

Furnham, A., & M. Argyle. (1998). *The psychology of money*. London ; New York: Routledge.

Gabriel, V. (2004). *Rechts- und Gerichtswesen im Lande Wursten vom Ausgang des Mittelalters bis ins 17. Jahrhundert*. E-Dissertation der Universität Hamburg (02. June). Retrieved 25. June, 2007, from http://www.sub.uni-hamburg.de/opus/volltexte/2004/2115/.

Galaxy Zoo. (2007). *The Galaxy Zoo welcomes a flood of visitors* (8. August). Retrieved 6. August, 2008, from http://www.galaxyzoo.org/pr_080807.aspx.

Galbraith, J.R. (1977). *Organization design*. Reading: Addison-Wesley.

Gallup, G., & S.F. Rae. (1940). Is there a bandwagon vote? *Public Opinion Quarterly, 4*(2), 244-249.

Galpin, V. (2002). Women in computing around the world. *ACM SIGCSE Bulletin, 34*(2), 94-100.

Gambardella, A., & B.H. Hall. (2006). Proprietary versus public domain licensing of software and research products. *Research Policy, 35*(6), 875-892.

Gambetta, D. (1988). Mafia: The price of distrust. In: D. Gambetta (Ed.), *Trust: Making and breaking cooperative relations* (pp. 158-175). New York: Basil Blackwell.

Gardner, P.L. (1975). Scales and statistics. *Review of Educational Research, 45*(1), 43-57.

Garfield, E. (1979). *Citation indexing its theory and application in science, technology, and humanities*. New York: Wiley.

Garrette, B., & B. Quelin. (1994). An empirical study of hybrid forms of governance structure: The case of the telecommunication equipment industry. *Research Policy, 23*(4), 395-412.

Gauck, J. (2000). Dankansprache. In: *Dolf Sternberger-Preis 2000: Reden bei der Überreichung des Preises an Joachim Gauck am 15. Juli 2000 in Erfurt* (pp. 46-55). Heidelberg: Dolf Sternberger-Gesellschaft.

Gautschi, T. (2000). History effects in social dilemma situations. *Rationality and Society, 12*(2), 131-162.

Gelade, G.A., P. Dobson, & P. Gilbert. (2006). National differences in organizational commitment: Effect of economy, product of personality, or consequence of culture? *Journal of Cross-Cultural Psychology, 37*(5), 542-556.

Gelernter, D.H. (1999). *Machine beauty: Elegance and the heart of technology* (Reprint ed.). New York: Basic Books.

General Accounting Office. (1992). *B-247094: Report to the House of Representatives*. GAO: Information Management and Technology Division (4. February). Retrieved 15. May, 2007, from http://www.fas.org/spp/starwars/gao/im92026.htm.

Gerwin, D. (1981). Relationship between structure and technology. In: P. C. Nystrom & W. H. Starbuck (Eds.), *Handbook of organizational design* (Vol. 2, pp. 3-38). Oxford: Oxford University Press.

GESIS. (2007). *Online-Befragungen: Software* (4. April). Retrieved 27. April, 2007, from http://www.gesis.org/Methodenberatung/Datenerhebung/Online/software.htm#freesoftware.

Ghosh, A., & G. Rüdiger. (2006). *FLOSSPOLS project: Skills survey interim report*. UNU-MERIT. Retrieved 22. January, 2007, from http://flosspols.org/deliverables/FLOSSPOLS-D10-skills%20survey_interim_report-revision-FINAL.pdf.

Ghosh, R., & V.V. Prakash. (2000). The Orbiten free software survey. *First Monday, 5*(7).

Ghosh, R.A. (2005). Understanding free software developers: Findings from the FLOSS study. In: J. Feller, B. Fitzgerald, S. Hissam & K. R. Lakhani (Eds.), *Perspectives on free and open source software* (pp. 23-45). Cambridge: MIT Press.

Ghosh, R.A., R. Glott, B. Krieger, & G. Robles. (2002a). *Floss final report - Part 4: Survey of developers*. Maastricht: International Institute of Infonomics, University of Maastricht.

Ghosh, R.A., R. Glott, B. Krieger, & G. Robles. (2002b). *Floss final report - Part 4A: Survey of developers - Annexure on validation and methodology*. Maastricht: International Institute of Infonomics, University of Maastricht.

Ghosh, R.A., R. Glott, B. Krieger, & G. Robles. (2002c). *Floss final report - Survey of developers: Raw data and further information/comments*. Maastricht: International Institute of Infonomics, University of Maastricht.

Ghoshal, S., & P. Moran. (1996). Bad for practice: A critique of the transaction cost theory. *Academy of Management Review, 21*(1), 13-47.

Gibbons, J.D. (1985). *Nonparametric methods for quantitative analysis* (2 ed.). Columbus: American Sciences Press.

Gibbons, R. (1992). *A primer in game theory*. New York: Harvester.

Gigerenzer, G. (2002). *Calculated risks: How to know when numbers deceive you*. New York: Simon & Schuster.

Giles, J. (2007). Key biology databases go wiki. *Nature, 445*(7129), 691.

Gilliland, S.W., & D. Chan. (2001). Justice in organizations: Theory, methods, and applications. In: N. Anderson, D. S. Ones, H. K. Sinangil & C. Viswesvaran (Eds.), *Handbook of industrial, work and organizational psychology* (Vol. 2, pp. 143-165). London: Sage.

Gintis, H. (2000). *Game theory evolving: A problem-centered introduction to modeling strategic behaviour*. Princeton: Princeton University Press.

Gladwell, M. (2000). *The tipping point: How little things can make a big difference* (1 ed.). Boston: Little Brown.

Glass, R.L. (2005). "Silver bullet" milestones in software history. *Communications of the ACM, 48*(8), 15-18.

Gleick, P.H. (2004). The myth and reality of bottled water. In: P. H. Gleick & Pacific Institute (Eds.), *The world's water : the biennial report on freshwater resources 2004-2005* (pp. 17-43). Washington: Island Press.

Glenn, N.D. (1976). Cohort analysts futile quest: Statistical attempts to separate age, period and cohort effects. *American Sociological Review, 41*(5), 900-904.

Goetz, T. (2003). Open source everywhere. *Wired, 11*(11).

Gokhale, J., L.J. Kotlikoff, & J. Sabelhaus. (1996). Understanding the postwar decline in US saving: A cohort analysis. *Brookings Papers on Economic Activity*(1), 315-407.

Gold, V. (2004). *Turing Award honors Alan Kay as pioneer of personal computers*. ACM Press Release 19. April. Retrieved 18. October, 2006, from http://www.acm.org/announcements/turingaward.4-21-2004.html.

Goldman, R., & R.P. Gabriel. (2005). *Innovation happens elsewhere: Open source as business strategy*. Amsterdam: Elsevier.

Goldstone, J.A. (1980). Review: Theories of revolution - The third generation. *World Politics, 32*(3), 425-453.

Google. (2007). *Google jobs: What's it like to work in engineering, operations, & IT?* Retrieved 8. January, 2007, from http://www.google.com/support/jobs/bin/static.py?page=about.html.

Gordon, H.S. (1954). The economic theory of a common property resource: The fishery. *Journal of Political Economy, 62*(2), 124-142.

Goyal, A. (1996). Women in computing: Historical roles, the perpetual glass ceiling, and current opportunities. *IEEE Annals of the History of Computing, 18*(3), 36-42.

Gracyk, T., & F.W. Hoffmann. (2000). *Popular American recording pioneers: 1895-1925*. New York: Haworth Press.

Graham, P. (2005). *Web 2.0*. Retrieved 4. December, 2006, from http://www.paulgraham.com/web20.html.

Granovetter, M. (1978). Threshold models of collective behavior. *American Journal of Sociology, 83*(6), 1420-1443.

Granovetter, M. (1985). Economic action and social structure: The problem of embeddedness. *American Journal of Sociology, 91*(3), 481-510.

Grant, M.J., & V. Sermat. (1969). Status and sex of other as determinants of behavior in a mixed motive game. *Journal of Personality and Social Psychology, 12*(2), 151-&.

Gray, J. (1985). Why do computers stop and what can be done about it? *Tandem Technical Report, 85.7*, 1-35.

Greatrex, N. (2004). *What is a scab?* BBC News. Retrieved 19. December, 2006, from http://news.bbc.co.uk/2/hi/uk_news/3437083.stm.

Green, G.I. (1989). Perceived importance of systems analysts job skills, roles, and non-salary incentives. *MIS Quarterly, 13*(2), 115-133.

Greenberg, J. (1990). Organizational justice: Yesterday, today, and tomorrow. *Journal of Management, 16*(2), 399-432.

Greenberg, J. (1997). A social influence model of employee theft: Beyond the fraud triangle. In: R. J. E. Lewicki, R. J. E. Bies & B. H. E. Sheppard (Eds.), *Research on negotiation in organizations* (pp. 29-51): JAI Press.

Greiner, L.E. (1972). Evolution and revolution as organizations grow. *Harvard Business Review, 50*(4), 37-46.

Greiner, L.E. (1998). Evolution and revolution as organizations grow. *Harvard Business Review, 76*(3), 55-67.

Grimm, J., & W. Grimm. (1812). *Kinder und Hausmärchen* (Vol. Grosse Ausgabe Band 1). Göttingen: Verlag der Dieterichschen Buchhandlung.

Grindley, P.C., & D.J. Teece. (1997). Managing intellectual capital: Licensing and cross-licensing in semiconductors and electronics. *California Management Review, 39*(2), 8-41.

Grinter, R.E., J.D. Herbsleb, & D.E. Perry. (1999, 14.-17. November). *The geography of coordination: Dealing with distance in R&D work.* Paper presented at the International ACM SIGGROUP Conference on Supporting Group Work, Phoenix.

Grodzins, M. (1958). *The metropolitan area as a racial problem*. Pittsburgh: University of Pittsburgh Press.

Groklaw. (2003). *The GPL is a license, not a contract, which is why the sky isn't falling*. Retrieved 26. June, 2006, from http://www.groklaw.net/article.php?story=20031214210634851.

Groklaw. (2006a). *Jeremy Allison has resigned from Novell to protest MS patent deal*. Retrieved 8. January, 2007, from http://www.groklaw.net/article.php?story=20061221081000710.

Groklaw. (2006b). *Kimball rules: SCO's objections denied! Wells's order affirmed: Novell goes 1st*. Retrieved 22. January, 2007, from http://www.groklaw.net/article.php?story=20061129165103775.

Groklaw. (2006c). *The Novell-Microsoft patent agreement timeline: Resource page*. Retrieved 8. January, 2007, from http://www.groklaw.net/staticpages/index.php?page=20061218045851480.

Grotius, H. (1978). *Mare liberum sive de iure quod Batavis competit ad Indicana Commercia dissertatio* (Vol. 1). Osnabrück: Biblio Verlag.

Grottke, M., & K.S. Trivedi. (2007). Fighting bugs: Remove, retry, replicate, and rejuvenate. *Computer, 40*(2), 107-109.

Grow, B. (2007). *Out at Home Depot: Behind the flameout of controversial CEO Bob Nardelli*. Business Week Online (15. January). Retrieved 16. January, 2007, from http://www.businessweek.com/magazine/content/07_03/b4017001.htm.

Gurr, T.R., & Woodrow Wilson School of Public and International Affairs (Center of International Studies). (1970). *Why men rebel*. Princeton: Published for the Center of International Studies Princeton University by Princeton University Press.

Gustafsson, M., A. Biel, & T. Garling. (1999). Overharvesting of resources of unknown size. *Acta Psychologica, 103*(1-2), 47-64.

Guth, W., R. Schmittberger, & B. Schwarze. (1982). An experimental analysis of ultimatum bargaining. *Journal of Economic Behavior & Organization, 3*(4), 367-388.

Guttman, J.M. (1950). The basis of scalogram analysis. In: S. A. Stouffer, J. M. Guttman, E. A. Suchman, P. F. Lazarsfeld, S. A. Star & J. A. Clausen (Eds.), *Studies in social psychology in World War II: Measurement and Prediction* (Vol. 4, pp. 60-90). Princeton: Princeton University Press.

Gysi, G. (1999). *Ueber Gott und die Welt: Gregor Gysi im Gespräch mit Daniela Dahn, Lothar de Maizière, Hans-Otto Bräutigam und Lothar Bisky*. Berlin: Schwarzkopf & Schwarzkopf.

Hackman, J.R., & G.R. Oldham. (1975). Development of job diagnostic survey. *Journal of Applied Psychology, 60*(2), 159-170.

Hackman, J.R., & G.R. Oldham. (1976). Motivation through design of work: Test of a theory. *Organizational Behavior and Human Performance, 16*(2), 250-279.

Haefliger, S., G. von Krogh, & S. Spaeth. (2007). Code reuse in open source software. *Management Science, Forthcoming*, 1-34.

Hagedorn, H. (2003). *Patenting software and services*. OECD (OECD Conference IPR: Innovation and economic performance (28. - 29. August). Retrieved 12. February, 2007, from http://www.oecd.org/dataoecd/48/49/12600939.pdf.

Hall, B.H. (2003). Business method patents, innovation, and policy. *UC Berkeley Competition Policy Center Working Paper, CPC03-39* (4. May), http://papers.ssrn.com/sol3/papers.cfm?abstract_id=463160.

Hall, B.H., J.L. Graham, D. Harhoff, & D.C. Mowery. (2003). Prospects for improving U.S. patent quality via post-grant opposition. *NBER Working Paper Series, 9737*, 1-25.

Hall, D.J., & M.A. Saias. (1980). Strategy follows structure. *Strategic Management Journal, 1*(2), 149-163.

Hamburger, H., M. Guyer, & J. Fox. (1975). Group size and cooperation. *Journal of Conflict Resolution, 19*(3), 503-531.

Hamel, G. (2000). *Leading the revolution*. Boston: Harvard Business School Press.

Hamel, G. (2002). *Leading the revolution* (2 ed.). New York: Plume.

Hamel, G. (2006). The why, what, and how of management innovation. *Harvard Business Review, 84*(2), 72-92.

Hamilton, A., J. Madison, J. Jay, & J.R. Pole. (2005). *The federalist*. Indianapolis: Hackett Publishing Company.

Hamm, S. (2004). *Linus Torvalds' benevolent dictatorship*. Business Week online (18. August). Retrieved 21. September, 2006, from http://www.businessweek.com/print/technology/content/aug2004/tc20040818_1593.htm?tc.

Handy, C. (1995). Trust and the virtual organization. *Harvard Business Review, 73*(3), 40-50.

Hanfeld, M. (2007). *Der Amoklauf auf CNN: Da wurde ein Zeuge zum Kollegen*. FAZ (17. April). Retrieved 29. May, 2007, from http://www.faz.net/s/Rub5A6DAB001EA2420BAC082C25414D2760/Doc~EEFC16DD7ADAA47C18A40C1E639613051~ATpl~Ecommon~Scontent.html.

Hannum, C.M. (2006). *The future of NetBSD*. Retrieved 18. September, 2006, from http://mail-index.netbsd.org/netbsd-users/2006/08/30/0016.html.

Hansmann, H. (1990). When does worker ownership work: Esops, law firms, codetermination, and economic democracy. *Yale Law Journal, 99*(8), 1749-1816.

Hansmann, H.B. (1980). The role of nonprofit enterprise. *Yale Law Journal, 89*(5), 835-901.

Harackiewicz, J.M. (1979). Effects of reward contingency and performance feedback on intrinsic motivation. *Journal of Personality and Social Psychology, 37*(8), 1352-1363.

Harbaugh, W., K. Krause, & S. Liday. (2002). Bargaining by children. *University of Oregon Economics Department Working Papers, 2002-4*, 38.

Harcourt, B.E., & J. Ludwig. (2006). Broken windows: New evidence from New York and a five-city social experiment. *University of Chicago Law Review, 73*(1), 271-320.

Hardin, G. (1968). The tragedy of the commons. *Science, 162*(3859), 1243-1248.

Hardin, R. (1982). *Collective action*. Baltimore: Johns Hopkins University Press.

Hardmeier, S., & H. Roth. (2003). Researching the influence of public opinion polls: Drawing lessons from the Swiss 'case'. *Politische Vierteljahresschrift, 44*(2), 174-195.

Harhoff, D., J. Henkel, & E. von Hippel. (2003). Profiting from voluntary information spillovers: how users benefit by freely revealing their innovations. *Research Policy, 32*(10), 1753-1769.

Harrison, G.W., & J. Hirshleifer. (1989). An experimental evaluation of weakest link/best shot models of public goods. *Journal of Political Economy, 97*(1), 201-225.

Hars, A., & S.S. Ou. (2002). Working for free? Motivations for participating in open source projects. *International Journal of Electronic Commerce, 6*(3), 25-39.

Harsanyi, J.C. (1977). *Rational behavior and bargaining equilibrium in games and social situations*. Cambridge Eng. ; New York: Cambridge University Press.

Hartman, P., J.P. Bezos, S. Kaphan, & J. Spiegel. (1999). Method and system for placing a purchase order via a communications network. USPTO 5,960,411: Amazon.com (Inc.).

Hatton, L. (1997). Reexamining the fault density: Component size connection. *IEEE Software, 14*(2), 89-97.

Hauert, C., S. De Monte, J. Hofbauer, & K. Sigmund. (2002). Volunteering as Red Queen mechanism for cooperation in public goods games. *Science, 296*(5570), 1129-1132.

Hauser, M.D., M.K. Chen, F. Chen, E. Chuang, & E. Chuang. (2003). Give unto others: genetically unrelated cotton-top tamarin monkeys preferentially give food to those who altruistically give food back. *Proceedings of the Royal Society of London Series B-Biological Sciences, 270*(1531), 2363-2370.

Havel, V. (1985). The power of the powerless. In: V. Havel & J. Keane (Eds.), *The power of the powerless: Citizens against the state in Central-Eastern Europe* (pp. 23-96). Armonk: M.E. Sharpe.

Hayek, F.A. (1978). *Law, legislation and liberty: Rules and order* (Vol. 1). Chicago: University of Chicago Press.

Hayek, F.A.v. (1979). *The counter-revolution of science: Studies on the abuse of reason* (2 ed.). Glencoe, Ill.,: Free Press.

Head, J.G. (1977). Public goods: Polar case reconsidered. *Economic Record, 53*(142), 227-238.

Head, J.G., & C.S. Shoup. (1969). Public goods, private goods, and ambiguous goods. *Economic Journal, 79*(315), 567-572.

Hechter, M. (1984). When actors comply: Monitoring costs and the production of social order. *Acta Sociologica, 27*(3), 161-183.

Heckathorn, D.D. (1996). The dynamics and dilemmas of collective action. *American Sociological Review, 61*(2), 250-277.

Hecker, F. (1999). Setting up shop: The business of open source software. *IEEE Software, 16*(1), 45-51.

Heckhausen, H. (1989). *Motivation und Handeln* (2 ed.). Berlin: Springer.

Hedges, L.V., & I. Olkin. (1985). *Statistical methods for meta-analysis*. Orlando: Academic Press.

Heider, F. (1958). *The psychology of interpersonal relations*. New York: Wiley.

Heiman, G.W. (1995). *Research methods in psychology*. Boston: Houghton Mifflin.

Heinen, J.S., & E. Jacobson. (1976). A model of task group development in complex organizations and strategy of implementation. *Academy of Management Review, 1*(4), 98-111.

Heise online. (2006). *Mozilla Firefox gewinnt wieder Marktanteile* (11. June 2006). Retrieved 13. September, 2006, from http://www.heise.de/newsticker/meldung/75311.

Hellwig, C. (2002). [PATCH] make LSM register functions GPLonly exports. *Linux kernel mailing list, 17. October*.

Hemelrijk, C.K. (2005). *Self-organisation and evolution of social systems*. Cambridge: Cambridge University Press.

Hendry, J. (1987). The teashop computer manufacturer: J. Lyons, Leo and the potential and limits of high-tech diversification. *Business History, 29*(1), 73-102.

Henkel, J. (2006). Selective revealing in open innovation processes: The case of embedded Linux. *Research Policy, 35*(7), 953-969.

Henn, M. (1998). Opinion polling in Central and Eastern Europe under communism. *Journal of Contemporary History, 33*(2), 229-240.

Henrich, J., R. Boyd, S. Bowles, C. Camerer, E. Fehr, H. Gintis, et al. (2005). "Economic man" in cross-cultural perspective: Behavioral experiments in 15 small-scale societies. *Behavioral and Brain Sciences, 28*(6), 795-815.

Herbsleb, J.D., & R.E. Grinter. (1999). *Splitting the organization and integrating the code: Conway's law revisited.* Paper presented at the 21st International Conference on Software Engineering, Los Angeles.

Herbsleb, J.D., A. Mockus, T.A. Finholt, & R.E. Grinter. (2001). Distance, dependencies, and delay in a global collaboration. *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, 319-328.

Hertel, G., J. Neuhof, T. Theuer, & N.L. Kerr. (2000). Mood effects on cooperation in small groups: Does positive mood simply lead to more cooperation? *Cognition & Emotion, 14*(4), 441-472.

Hertel, G., S. Niedner, & S. Herrmann. (2003). Motivation of software developers in open source projects: an Internet-based survey of contributors to the Linux kernel. *Research Policy, 32*(7), 1159-1177.

Herzberg, F. (2003). One more time: How do you motivate employees? *Harvard Business Review, 81*(1), 87-+.

Hess, H. (2003). Like Zealots and Romans: Terrorism and empire in the 21st century. *Crime Law and Social Change, 39*(4), 339-357.

Hickson, D.J., D.S. Pugh, & D.C. Pheysey. (1969). Operations technology and organization structure: Empirical reappraisal. *Administrative Science Quarterly, 14*(3), 378-397.

Hiltz, S.R., K. Johnson, & M. Turoff. (1986). Experiments in group decision making: Communication process and outcome in face-to-face versus computerized conferences. *Human Communication Research, 13*(2), 225-252.

Himanen, P. (2001). *The hacker ethic, and the spirit of the information age* (1 ed.). New York: Random House.

Hinchcliffe, D. (2006). *The state of Web 2.0*. Retrieved 4. December, 2006, from http://web2.wsj2.com/the_state_of_web_20.htm.

Hirschman, A.O. (1970). *Exit, voice, and loyalty; responses to decline in firms, organizations, and states*. Cambridge, Mass.,: Harvard University Press.

Hirschman, A.O. (1993). Exit, voice, and the fate of the German Democratic Republic: An essay in conceptual history. *World Politics, 45*(2), 173-202.

Hirshleifer, J. (1963). *Disaster and recovery: A historical survey*. Santa Monica: Rand Corporation.

Hirshleifer, J. (1985). The expanding domain of economics. *American Economic Review, 75*(6), 53-68.

Hobbes, T. (1998). *Leviathan*. Oxford: Oxford University Press.

Hobcraft, J., J. Menken, & S. Preston. (1982). Age, period, and cohort effects in demography: A review. *Population Index, 48*(1), 4-43.

Hof, R.D. (2003). Tech outfits should take notes. *Business Week, 3. March*(3822), 86-86.

Hof, R.D. (2004). Something Wiki this way comes. *Business Week, 3886*(7. June), 128.

Hofstede, G. (1980). *Culture's consequences: International differences in work-related values*. Beverly Hills: SAGE.

Hofstede, G.H. (2001). *Culture's consequences: Comparing values, behaviors, institutions, and organizations across nations* (2 ed.). Thousand Oaks: Sage.

Hofstede, G.H., & G.J. Hofstede. (2005). *Cultures and organizations: Software of the mind*. New York: McGraw-Hill.

Hollander, M., & D.A. Wolfe. (1999). *Nonparametric statistical methods* (2 ed.). New York: J. Wiley.

Holmstrom, B. (1999). Managerial incentive problems: A dynamic perspective. *Review of Economic Studies, 66*(1), 169-182.

Holmström, B.R., & P. Milgrom. (1991). Multitask principal agent analyses: Incentive contracts, asset ownership, and job design. *Journal of Law, Economics and Organization, 7*(Special Issue), 24-52.

Home Depot. (2007). *News release: Robert Nardelli resigns as chairman and Chief Executive Officer of The Home Depot and is succeeded by Frank Blake* (3. January). Retrieved 16. January, 2007, from http://ir.homedepot.com/ReleaseDetail.cfm?ReleaseID=224078.

Hopfensitz, A., & E. Reuben. (2005). The importance of emotions for the effectiveness of social punishment. *Tinbergen Institute Discussion Paper, 2005-075/1*, 1-31.

Horai, J., & J.T. Tedeschi. (1975). Compliance and use of threats and promises after a power reversal. *Behavioral Science, 20*(2), 117-124.

Hornby, A.S. (2000). *Oxford advanced learner's dictionary of current English* (6 ed.). Oxford: Oxford University Press.

Horsley, R.A. (1986). The Zealots, their origin, relationships and importance in the Jewish revolt. *Novum Testamentum, 28*(2), 159-192.

Hotelling, H. (1929). Stability in competition. *The Economic Journal, 39*(153), 41-57.

Hottes, J.H., & A. Kahn. (1974). Sex differences in a mixed motive conflict situation. *Journal of Personality, 42*(2), 260-275.

House, R.J., & Global Leadership and Organizational Behavior Effectiveness Research Program. (2004). *Culture, leadership, and organizations: The GLOBE study of 62 societies*. Thousand Oaks: Sage.

Huber, M. (2006). Free software on the campus: FOSDEM 2006 in Brussels. *Linux Magazine, 66*(31. March), 90.

Hubmann, V. (2007). *Cura Vista - Database of the Swiss Parliament: Parlamentarier als Vorbild für die Bevölkerung?* Swiss Parliamentary Service, (04.5054, submitted in 03.03.2004). Retrieved 8. January, 2007, from http://www.parlament.ch/e/homepage/su-suchen/su-curia-vista.htm.

Huck, S.W., & N.D. Bowers. (1972). Item difficulty level and sequence effects in multiple choice achievement tests. *Journal of Educational Measurement, 9*(2), 105-111.

Hulin, C.L., & M.R. Blood. (1968). Job enlargement, individual differences, and worker responses. *Psychological Bulletin, 69*(1), 41-55.

Hulland, J. (1999). Use of partial least squares (PLS) in strategic management research: A review of four recent studies. *Strategic Management Journal, 20*(2), 195-204.

Human Genome Project. (2006). *Genetics and patenting*. Retrieved 4. December, 2006, from http://www.ornl.gov/sci/techresources/Human_Genome/elsi/patents.shtml#7.

Hume, D. (1739). *A treatise of human nature: Being an attempt to introduce the experimental method of reasoning into moral subjects*. London: Printed for J. Noon.

IAAF. (2007). *All records 100 metres men: World record* (28. February). Retrieved 28. February, 2007, from http://www.iaaf.org/statistics/records/gender=M/allrecords/discipline=100/index.html.

IBM. (2006). *IBM invests US$2.2 million to expand Linux technology center in Brazil*. IBM Press Release (23. May). Retrieved 25. June, 2007, from http://www-03.ibm.com/press/us/en/pressrelease/19697.wss.

IDC. (2006). *Worldwide server market experiences lackluster quarter across multiple segments*. IDC Press Release (24. May). Retrieved 25. June, 2007, from http://www.idc.com/getdoc.jsp?containerId=prUS20180706.

IDS St. Gallen. (2007). *Welcome to the library catalogue IDS St. Gallen* (search term "raymond" and "bazaar"). Retrieved 21. June, 2007, from http://aleph.unisg.ch/F?con_lng=eng.

IEEE Standards Board. (1989). IEEE standard for software reviews and audits. *ANSI / IEEE Standard 1028-1988, 30. June*, 1-35.

ifrOSS. (2006). *License Center*. Institut für Rechtsfragen der Open Source Software. Retrieved 15. June, 2006, from http://www.ifross.de/ifross_html/lizenzcenter-en.html.

ifrOSS, & T. Jaeger. (2005). *Die GPL kommentiert und erklärt: Rechtliche Erläuterungen zur GNU General Public License*. Köln: O'Reilly.

ILO. (2002). *Key indicators of the labour market: 2001-2002*. New York: Routledge.

Insko, C.A., J. Schopler, R.H. Hoyle, G.J. Dardis, & K.A. Graetz. (1990). Individual group discontinuity as a function of fear and greed. *Journal of Personality and Social Psychology, 58*(1), 68-79.

Isen, A.M. (1987). Positive affect, cognitive processes, and social behavior. *Advances in Experimental Social Psychology, 20*, 203-253.

Isen, A.M., & R.A. Baron. (1991). Positive affect as a factor in organizational behavior. *Research in Organizational Behavior, 13*, 1-53.

Isen, A.M., & J. Reeve. (2005). The influence of positive affect on intrinsic and extrinsic motivation: Facilitating enjoyment of play, responsible work behavior, and self-control. *Motivation and Emotion, 29*(4), 297-325.

ISO. (1998). *Overview of ISO/IEC JTC1/SC22/WG9* (4. December). Retrieved 2. March, 2007, from http://www.open-std.org/jtc1/sc22/wg9/overview.htm.

ISO. (2007). *ISO 3166 code lists*. Retrieved 22. June, 2007, from http://www.iso.ch/iso/en/prods-services/iso3166ma/02iso-3166-code-lists/index.html.

Jaccard, J., & R. Turrisi. (2003). *Interaction effects in multiple regression* (2 ed.). Thousand Oaks: Sage.

Jackson, T.P. (1999). *United States v. Microsoft (Civil Action No. 98-1232 and 98-1233): Court's findings of fact.* United States District Court for the District of Columbia. Retrieved 10. October, 2006, from http://www.usdoj.gov/atr/cases/f3800/msjudge.pdf.

Jakobs, K. (2006). Shaping user-side innovation through standardisation: The example of ICT. *Technological Forecasting and Social Change, 73*(1), 27-40.

James, B.G. (1974). Theory of corporate life cycle. *Long Range Planning, 7*(2), 49-55.

Jamgotch, N. (1985). Crisis communications and the hot line. In: N. Jamgotch (Ed.), *Sectors of mutual benefit in U.S.-Soviet relations* (pp. 8-21). Durham: Duke University Press.

Jang, K.L., R.R. McCrae, A. Angleitner, R. Riemann, & W.J. Livesley. (1998). Heritability of facet-level traits in a cross-cultural twin sample: Support for a hierarchical model of personality. *Journal of Personality and Social Psychology, 74*(6), 1556-1565.

Jap, S.D., & E. Anderson. (2007). Testing a life-cycle theory on cooperative interorganizational relationships: Movement across stages and performance. *Management Science, 53*(2), 260-275.

Jarvis, C.B., S.B. MacKenzie, & P.M. Podsakoff. (2003). A critical review of construct indicators and measurement model misspecification in marketing and consumer research. *Journal of Consumer Research, 30*(2), 199-218.

Jaspert, J. (2006). *Position statement to the Dunc-Tanc "experiment"*. Mailing list "debian-devel-announce@lists.debian.org" (26. October). Retrieved 3. January, 2007, from http://lists.debian.org/debian-project/2006/10/msg00260.html.

Jawahar, I.M., & G.L. McLaughlin. (2001). Toward a descriptive stakeholder theory: An organizational life cycle approach. *Academy of Management Review, 26*(3), 397-414.

Jefferson, T. (1813). *The Thomas Jefferson Papers Series. Series 1 - General correspondence 1651-1827: Thomas Jefferson to Isaac McPherson 13. August*. Library of Congress. Retrieved 14. February, 2007, from http://memory.loc.gov/ammem/collections/jefferson_papers/mtjser1.html.

Jeffreys, H., & B.S. Jeffreys. (2000). *Methods of mathematical physics* (3 ed.). Cambridge: Cambridge University Press.

Jeffries, R. (2004). *Extreme programming adventures in C#*. Redmond: Microsoft Press.

Jenkins, E. (2007). *Honored in the breach: Brush up your Shakespeare, act I.* Columbia Journalism Review. Retrieved 19. June, 20007, from http://www.cjr.org/resources/lc/honored.php.

Jensen, R., & N. Miller. (2002). Giffen behavior: Theory and evidence. *Harvard University Faculty Research Working Papers Series, RWP02-014*, 28.

Johannessen, J.A., J. Olaisen, & B. Olsen. (1997). Information management in negotiations: The conditions under which it could be expected that the negotiation partners substitute a competitive definition of the situation for a cooperative one. *International Journal of Information Management, 17*(3), 153-168.

Johnson, M. (2005). Wicksell's unanimity rule: Buchanan's dominance considered. *American Journal of Economics and Sociology, 64*(4), 1049-1071.

Johnson, T., P. Kulesa, Y.I. Cho, & S. Shavitt. (2005). The relation between culture and response styles: Evidence from 19 countries. *Journal of Cross-Cultural Psychology, 36*(2), 264-277.

Jones, J. (2005a). The Sinatra doctrine. *Harvard International Review, 27*(2), 80-81.

Jones, P. (2005b). *A look at the Patent Commons Project and OIN*. Linux Weekly News (7. December). Retrieved 12. February, 2007, from http://lwn.net/Articles/163038/.

Jones, W.H. (1979). Generalizing mail survey inducement methods: Population interactions with anonymity and sponsorship. *Public Opinion Quarterly, 43*(1), 102-111.

Joomla! (2007). *Joomla! community forum: Statistics center* (9. February). Retrieved 9. February, 2007, from http://forum.joomla.org/index.php?action=stats.

Jorgensen, N. (2000). *Results of the FreeBSD survey*. Retrieved 22. September, 2006, from http://webhotel.ruc.dk/nielsj//research/freebsd/freebsd.html.

Jorgensen, N. (2001). Putting it all in the trunk: Incremental software development in the FreeBSD open source project. *Information Systems Journal, 11*(4), 321-336.

Jorgensen, N. (2005). Incremental and Decentralized Integration in FreeBSD. In: J. Feller, B. Fitzgerald, S. Hissam & K. R. Lakhani (Eds.), *Perspectives on free and open source software* (pp. 227-243). Cambridge: MIT Press.

Judge, T.A., S. Parker, A.E.H. Colbert, D., & R. Ilies. (2001). Job satisfaction: A cross cultural review. In: N. Anderson, D. S. Ones, H. K. Sinangil & C. Viswesvaran (Eds.), *Handbook of industrial, work and organizational psychology* (Vol. 2, pp. 25-52). London: Sage.

Julious, S.A., & M.A. Mullee. (1994). Confounding and Simpsons paradox. *British Medical Journal, 309*(6967), 1480-1481.

Kahn, J.M., A.A. Kramer, & G.D. Bubenfeld. (2007). Transferring critically ill patients out of hospital improves the standardized mortality ratio. *Chest, 13*(1), 68-75.

Kahneman, D., & A. Tversky. (1979). Prospect theory: Analysis of decision under risk. *Econometrica, 47*(2), 263-291.

Kaiser, A. (2007). *Augenzeuge des Amoklaufs: Der Mann mit dem Kamerahandy*. FAZ (17. April). Retrieved 29. May, 2007, from http://www.faz.net/s/Rub77CAECAE94D7431F9EACD163751D4CFD/Doc~E2B2B8F37B6DF4E27AA52F434F187B235~ATpl~Ecommon~Scontent.html.

Kaiser, H.F. (1979). Citation classic: Varimax criterion for analytic rotation in factor-analysis. *Current Contents/Social & Behavioral Sciences, 24*, 33.

Kallenbach, G., & W. Schade. (1998). *Cold war: The wall comes down (Episode 23) - Episode script*. CNN. Retrieved 28. March, 2007, from http://www.cnn.com/SPECIALS/cold.war/episodes/23/script.html.

Kalman, M.E., P. Monge, J. Fulk, & R. Heino. (2002). Motivations to resolve communication dilemmas in database-mediated collaboration. *Communication Research, 29*(2), 125-154.

Kant, I. (1995). *Werke in sechs Bänden: Grundlegung zur Metaphysik der Sitten* (Vol. 3). Köln: Könemann.

Karau, S.J., & K.D. Williams. (1993). Social Loafing: A metaanalytic review and theoretical integration. *Journal of Personality and Social Psychology, 65*(4), 681-706.

Kato, A. (2004). Patent pool enhances market competition. *International Review of Law and Economics, 24*(2), 255-268.

Katz, D., & R.L. Kahn. (1978). *The social psychology of organizations* (2 ed.). New York: Wiley.

Katz, M.L., & C. Shapiro. (1985). Network externalities, competition, and compatibility. *American Economic Review, 75*(3), 424-440.

Kaufmann, K. (2002). *Zealots*. Jewish Encyclopedia. Retrieved 10. June, 2007, from http://www.jewishencyclopedia.com/view.jsp?artid=49&letter=Z&search=zealot.

KDE. (2007). *Credits* (9. March). Retrieved 9. March, 2007, from http://www.kde.org/people/credits/.

Kearns, W.E. (1937). *The speed of grizzly bears*. United States Department of the Interior: National Park Service (Yellowstone Nature Notes, 14(Nos. 1-2)). Retrieved 28. February, 2007, from http://www.cr.nps.gov/history/online_books/yell/vol14-1-2a.htm.

Kelly, C. (2005). Free Science. In: J. Feller, B. Fitzgerald, S. Hissam & K. R. Lakhani (Eds.), *Perspectives on free and open source software* (pp. 415-430). Cambridge: MIT Press.

Kennedy, B. (2007). *The birth of the hot line*. CNN interactive. Retrieved 7. March, 2007, from http://www.cnn.com/SPECIALS/cold.war/episodes/10/spotlight/.

Kernighan, B.W., & R. Pike. (1999). *The practice of programming*. Reading: Addison-Wesley.

Kernighan, B.W., & P.J. Plauger. (1976). *Software tools*. Reading: Addison-Wesley.

Kernighan, B.W., & P.J. Plauger. (1978). *The elements of programming style* (2nd ed.). New York: McGraw-Hill.

Kerr, N.L. (1989). Illusions of efficacy: The effects of group-size on perceived efficacy in social dilemmas. *Journal of Experimental Social Psychology, 25*(4), 287-313.

Kerr, N.L. (1995). Norms in social dilemmas. In: D. A. Schroeder (Ed.), *Social dilemmas: Perspectives on individuals and groups* (pp. 31-47). Westport: Praeger.

Kerr, S., & J.W. Slocum. (1981). Controlling the performances of people in organizations. In: P. C. Nystrom & W. H. Starbuck (Eds.), *Handbook of organizational design* (Vol. 2, pp. 116-133). Oxford: Oxford University Press.

Kerstetter, J. (2003). *Sun: It's not "Linux or nothing"*. Business Week Online. Retrieved 19. Mai, 2006, from http://www.businessweek.com/magazine/content/03_09/b3822614_tc102.htm.

Keser, C., & F. van Winden. (2000). Conditional cooperation and voluntary contributions to public goods. *Scandinavian Journal of Economics, 102*(1), 23-39.

Kessel, R.A. (1974). Transfused blood, serum hepatitis, and Coase theorem. *Journal of Law & Economics, 17*(2), 265-289.

Keynes, J.M. (1936). *The general theory of employment, interest and money*. London: Macmillan.

Kieser, A. (1987). From asceticism to administration of wealth: Medieval monasteries and the pitfalls of rationalization. *Organization Studies, 8*(2), 103-123.

Kiesler, S., D. Zubrow, A. Moses, & V. Geller. (1985). Affect in computer mediated communication: An experiment in synchronous terminal-to-terminal discussion. *Human Computer Interaction, 1*(1), 77-104.

Kim, W.C., & R. Mauborgne. (1997). Fair process: Managing in the knowledge economy. *Harvard Business Review, 75*(4), 65-75.

Kimball, A.W. (1957). Errors of the third kind in statistical consulting. *Journal of the American Statistical Association, 52*(278), 133-142.

Kimberly, J.R. (1979). Issues in the creation of organizations: Initiation, innovation, and institutionalization. *Academy of Management Journal, 22*(3), 437-457.

Kimmel, M.J., D.G. Pruitt, J.M. Magenau, E. Konargoldband, & P.J.D. Carnevale. (1980). Effects of trust, aspiration, and gender on negotiation tactics. *Journal of Personality and Social Psychology, 38*(1), 9-22.

Kirchgässner, G. (1992). Towards a theory of low-cost decisions. *European Journal of Political Economy, 8*(2), 305-320.

Kirchgässner, G. (1998). Bedingungen moralischen Handelns. *Working Paper of the University of St. Gallen (Department of Economics), 9808*, 1-20.

Kittur, A., B.A. Pendleton, B. Suh, & T. Mytkowicz. (2007). *Power of the few vs. wisdom of the crowd: Wikipedia and the rise of the bourgeoisie* (31. January). Retrieved 2. March, 2007, from http://www.viktoria.se/altchi/index.php?action=showsubmission&id=41.

Klandermans, B. (1997). *The social psychology of protest*. Oxford: Blackwell Publishers.

Klatsky, A.L., M.A. Armstrong, & H. Kipp. (1990). Correlates of alcoholic beverage preference: Traits of persons who choose wine, liquor or beer. *British Journal of Addiction, 85*(10), 1279-1289.

Klemperer, P. (1995). Competition when consumers have switching costs: An overview with applications to industrial organization, macroeconomics, and international trade. *Review of Economic Studies, 62*(4), 515-539.

Knapp, T.R. (1990). Treating ordinal scales as interval scales: An attempt to resolve the controversy. *Nursing Research, 39*(2), 121-123.

Knight, F.H. (1921). *Risk, uncertainty and profit*. Boston: Houghton Mifflin Company.

Knorr Cetina, K., & U. Bruegger. (2002). Global microstructures: The virtual societies of financial markets. *American Journal of Sociology, 107*(4), 905-950.

Knuth, D. (2002). All questions answered. *Notices of the AMS, 49*(3), 318-324.

Knuth, D. (2006). *Email (let's drop the hyphen)*. Retrieved 07. July, 2006, from http://www-cs-faculty.stanford.edu/~knuth/email.html.

Knuth, D.E. (1997). *The art of computer programming* (Vol. 1-3). Reading: Addison-Wesely.

Koestner, R., R.M. Ryan, F. Bernieri, & K. Holt. (1984). Setting limits on children’s behavior: The differential effects of controlling versus informational styles on intrinsic motivation and creativity. *Journal of Personality, 52*(3), 233-248.

Kollock, P. (1998). Social dilemmas: The anatomy of cooperation. *Annual Review of Sociology, 24*, 183-214.

Komorita, S.S., & C.W. Lapworth. (1982). Cooperative choice among individuals versus groups in an N-person dilemma situation. *Journal of Personality and Social Psychology, 42*(3), 487-496.

Komorita, S.S., & C.D. Parks. (1995). Interpersonal relations: Mixed motive interaction. *Annual Review of Psychology, 46*(1), 183-207.

Kondo, T. (1990). Some notes on rational behavior, normative behavior, moral behavior, and cooperation. *Journal of Conflict Resolution, 34*(3), 495-530.

Kopelman, S., M. Weber, & D. Messick. (2002). Factors influencing cooperation in commons dilemmas: A review of experimental psychological research. In: National Research Council (U.S.): Committee on the Human Dimensions of Global Change (Ed.), *The drama of the commons* (pp. 113-156). Washington: National Academy Press.

Kosiol, E. (1962). *Organisation der Unternehmung*. Wiesbaden: Gabler.

Kotrlik, J.W., & H.A. Williams. (2003). The incorporation of effect size in information technology, learning, and performance research. *Information Technology, Learning, and Performance Journal, 21*(1), 1-7.

Kotter, J.P. (1996). *Leading change*. Boston: Harvard Business School Press.

Kramer, R.M., & M.B. Brewer. (1986). Social group identity and the emergence of cooperation in resource conservation dilemmas. In: H. A. M. Wilke, D. M. Messick & C. G. Rutte (Eds.), *Experimental social dilemmas* (pp. 117-203). Frankfurt am Main: P. Lang.

Kramer, R.M., C.G. Mcclintock, & D.M. Messick. (1986). Social values and cooperative response to a simulated resource conservation crisis. *Journal of Personality, 54*(3), 576-592.

Kraut, R.E., & L.A. Streeter. (1995). Coordination in software development. *Communications of the ACM, 38*(3), 69-81.

Kreps, D.M. (1997). Intrinsic motivation and extrinsic incentives. *American Economic Review, 87*(2), 359-364.

Krishnamurthy, S. (2002). Cave or community? An empirical examination of 100 mature open source projects. *First Monday, 7*(6).

Kruskal, W.H., & W.A. Wallis. (1952). Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association, 47*(260), 583-621.

Kuhlman, D.M., & A.F.J. Marshello. (1975). Individual differences in game motivation as moderators of preprogrammed strategy effects in prisoners dilemma. *Journal of Personality and Social Psychology, 32*(5), 922-931.

Kuhn, B.M., & R. Stallman. (2006). *Freedom or power?* Free Software Foundation. Retrieved 13. June, 2005, from http://www.gnu.org/philosophy/freedom-or-power.html.

Kuhn, H.W. (1950). Simplified two person poker. In: H. W. Kuhn & A. W. Tucker (Eds.), *Contributions to the theory of games* (Vol. 1, pp. 97-103). Princeton, N.J.: Princeton University Press.

Kuhn, H.W. (1953). Extensive games and the problem of information. In: K. J. Arrow, E. W. Barankin & D. Blackwell (Eds.), *Contributions to the theory of games* (Vol. 2, pp. 193-216). Princeton: Princeton University Press.

Kunkel, R.G. (2002). Recent developments in shrinkwrap, clickwrap and browsewrap licenses in the United States. *Murdoch University Electronic Journal of Law, 9*(3), http://www.murdoch.edu.au/elaw/issues/v9n3/kunkel93_text.html.

Kunze, M. (1998). Lass es leuchten - LAMP: Ein datenbankgestütztes Web-Publishing-System mit Freeware. *c't, 98*(12), 230-231.

Kuran, T. (1989). Sparks and prairie fires: A theory of unanticipated political revolution. *Public Choice, 61*(1), 41-74.

Kuran, T. (1991). Now out of never: The element of surprise in the East European revolution of 1989. *World Politics, 44*(1), 7-48.

Kuran, T. (1995). *Private truths, public lies: The social consequences of preference falsification*. Cambridge: Harvard University Press.

Kuster, B. (1999). Exit, voice, and the theory of Anthony Downs: The power of extreme voters. *Contribution to the Fifth Annual General Social Survey Student Paper Competition, University of Chicago (NORC)*, 1-39.

Kuster, B. (2006). Vertrauen in geographisch verteilten Gruppen: Die Hudson's Bay Company und Open Source Produktion. In: M. Osterloh & A. Weibel (Eds.), *Investition Vertrauen: Prozesse der Vertrauensentwicklung in Organisationen* (pp. 207-239). Wiesbaden: Gabler.

Kuznets, S. (1940). Schumpeter's business cycles. *American Economic Review, 30*(2, Part 1), 257-271.

Kyng, M. (1991). Designing for cooperation: Cooperating in design. *Communications of the ACM, 34*(12), 64-73.

La Guardia, J.G., R.M. Ryan, C.E. Couchman, & E.L. Deci. (2000). Within-person variation in security of attachment: A self-determination theory perspective on attachment, need fulfillment, and well-being. *Journal of Personality and Social Psychology, 79*(3), 367-384.

Labovitz, S. (1967). Some observations on measurement and statistics. *Social Forces, 46*(2), 151-160.

Lado, A.A., & M.C. Wilson. (1994). Human resource systems and sustained competitive advantage: A competency-based perspective. *Academy of Management Review, 19*(4), 699-727.

Lakhani, K.R., B. Wolf, J. Bates, & C. DiBona. (2002). *The Boston Consulting Group hacker survey*. O'Reilly Open Source Conference 24. July (0.73). Retrieved 6. February, 2007, from http://www.ostg.com/bcg/bcg-0.73/BCGHackerSurveyv0-73.html.

Lakhani, K.R., & R.G. Wolf. (2005). Why hackers do what they do: Understanding motivation and effort in free/open source software projects. In: J. Feller, B. Fitzgerald, S. Hissam & K. R. Lakhani (Eds.), *Perspectives on free and open source software* (pp. 3-21). Cambridge: MIT Press.

Land, F. (2000). The first business computer: A case study in user-driven innovation. *IEEE Annals of the History of Computing, 22*(3), 16-26.

Landgericht München I. (2004). Verfügung 21 0 6123/04: Landgericht München I.

Lawler, E.E. (1971). *Pay and organizational effectiveness: A psychological view*. New York: McGraw-Hill.

Lawler, E.E. (2000). *Rewarding excellence: Pay strategies for the new economy* (1 ed.). San Francisco: Jossey-Bass.

Lawrence, P.R., & J.W. Lorsch. (1967). *Organization and environment; managing differentiation and integration*. Boston,: Division of Research, Graduate School of Business Administration, Harvard University.

Lazear, E.P. (2000). Performance pay and productivity. *American Economic Review, 90*(5), 1346-1361.

Lea, S.E.G., R.M. Tarpy, & P. Webley. (1987). *The individual in the economy: A textbook of economic psychology*. Cambridge: Cambridge University Press.

Lee, S., D. McCann, & J.C. Messenger. (2007). *Working time around the world*. London: Routledge.

Lee, S., N. Moisa, & M. Weiss. (2004). Conditions for open source as a signalling device. *Working Paper Series in Finance and Accounting of the Johann Wolfgang Goethe Universität Frankfurt, 102r*, 0-33.

Leibenstein, H. (1950). Bandwagon, snob, and Veblen effects in the theory of consumers' demand. *The Quarterly Journal of Economics, 64*(2), 183-207.

Leinonen, R., F.G. Diez, D. Binns, W. Fleischmann, R. Lopez, & R. Apweiler. (2004). UniProt archive. *Bioinformatics, 20*(17), 3236-3237.

Lenz, S. (1960). *Das Feuerschiff*. Hamburg: Hoffmann & Campe.

Leppämäki, M., & M. Mustonen. (2004). Signaling and screening with open source programming. *Helsinki School of Economics Working Paper, W-377*, 1-23.

Lerner, J., & J. Tirole. (2002). Some simple economics of open source. *Journal of Industrial Economics, 50*(2), 197-234.

Lerner, J., & J. Tirole. (2005). Economic perspective on open source. In: J. Feller, B. Fitzgerald, S. Hissam & K. R. Lakhani (Eds.), *Perspectives on free and open source software* (pp. 47-78). Cambridge: MIT Press.

Lerner, J., J. Tirole, & National Bureau of Economic Research. (2000). *The simple economics of open source*. Cambridge: National Bureau of Economic Research.

Leventhal, G.S. (1980). What should be done with equity theory? New approaches to the study of fairness in social relationships. In: K. Gergen, M. Greenberg & R. Willis (Eds.), *Social exchange: Advances in theory and research.* (pp. 27-55). New York: Plenum Press.

Leveson, N.G., & C.S. Turner. (1993). An investigation of the Therac-25 accidents. *Computer, 26*(7), 18-41.

Levitt, S.D., & S.J. Dubner. (2005). *Freakonomics: A rogue economist explores the hidden side of everything.* New York: Morrow.

Levy, S. (1984). *Hackers: Heroes of the computer revolution* (1 ed.). Garden City: Doubleday.

Liberman, V., S.M. Samuels, & L. Ross. (2004). The name of the game: Predictive power of reputations versus situational labels in determining prisoner's dilemma game moves. *Personality and Social Psychology Bulletin, 30*(9), 1175-1185.

Lieberman, M.B., & D.B. Montgomery. (1998). First mover (dis)advantages: Retrospective and link with the resource based view. *Strategic Management Journal, 19*(12), 1111-1125.

Liebowitz, S.J., & S.E. Margolis. (1990). The fable of the keys. *Journal of Law & Economics, 33*(1), 1-25.

Liebowitz, S.J., & S.E. Margolis. (1994). Network externality: An uncommon tragedy. *Journal of Economic Perspectives, 8*(2), 133-150.

Liebowitz, S.J., & S.E. Margolis. (1995). Path dependence, lock-in, and history. *Journal of Law Economics & Organization, 11*(1), 205-226.

Liebowitz, S.J., & S.E. Margolis. (2005). Seventeen famous economists weigh in on copyright: The role of theory, empirics, & network effects *Harvard Journal of Law & Technology, 18*(2), 435-457.

Liebrand, W.B.G. (1984). The effect of social motives, communication and group size on behavior in an N-person multi stage mixed motive game. *European Journal of Social Psychology, 14*(3), 239-264.

Liebrand, W.B.G., & G.J. Van Run. (1985). The effects of social motives on behavior in social dilemmas in two cultures. *Journal of Experimental Social Psychology, 21*(1), 86-102.

Lienert, G.A., & U. Raatz. (1998). *Testaufbau und Testanalyse* (6 ed.). Weinheim: Psychologie VerlagsUnion.

Likert, R. (1932). A technique for the measurement of attitudes. *Archives of Psychology, 140*, 4-55.

Lim, W.C. (1994). Effects of reuse on quality, productivity, and economics. *IEEE Software, 11*(5), 23-30.

Lind, E.A., J. Greenberg, K.S. Scott, & T.D. Welchans. (2000). The winding road from employee to complainant: Situational and psychological determinants of wrongful-termination claims. *Administrative Science Quarterly, 45*(3), 557-590.

Lind, E.A., & T.R. Tyler. (1988). *The social psychology of procedural justice*. New York: Plenum.

Lindenberg, S. (2001). Intrinsic motivation in a new light. *Kyklos, 54*(2-3), 317-343.

Lindsey, R., & R. Simmon. (2002). *Locust!* NASA Earth Observatory. Retrieved 9. January, 2007, from http://earthobservatory.nasa.gov/Study/Locusts/locusts3.html.

Linux Format. (2005). *Hack daddy*. Linux Format (69). Retrieved August 8, 2006, from http://www.linuxformat.co.uk/modules.php?op=modload&name=Sections&file=index&req=viewarticle&artid=15.

Linux News Desk. (2005). *LinuxWorld San Francisco 2005: LinuxWorld product excellence awards presented in SF*. Retrieved 15. November, 2006, from http://linux.sys-con.com/read/117705.htm.

Linux Weekly News. (2005). *Version control with GNU Arch*. Linux Weekly News (March 2). Retrieved 26. June, 2006, from http://lwn.net/Articles/125792/.

Linux Weekly News. (2006). *Harald Welte on the flood of GPL violations*. Linux Weekly News (June 22). Retrieved 26. June, 2006, from http://lwn.net/Articles/186944/.

Lippitt, G.L., & W.H. Schmidt. (1967). Crises in a developing organization. *Harvard Business Review, 45*(6), 102-112.

Littlewood, J.E. (1914). Sur la distribution des nombres premiers. *Comptes Rendus, 158*, 1869-1872.

Litwin, M.S. (1995). *How to measure survey reliability and validity* (Vol. 7). Thousand Oaks: Sage.

Liveright, M. (2006). *I will purchase the $100 laptop at $300 but only if 100,000 other will do the same*. PledgeBank (4. November and 31. October). Retrieved 27. February, 2007, from http://www.pledgebank.com/100laptop.

Locke, J. (1690). *Two treatises of government: In the former, the false principles, and foundation of Sir Robert Filmer, and his followers, are detected and overthrown. The latter is an essay concerning the true original, extent, and the end of civil government*. London,: A. Churchill.

Loevinger, J. (1954). The attenuation paradox in test theory. *Psychological Bulletin, 51*(5), 493-504.

Lohmann, S. (1994). The dynamics of informational cascades: The Monday Demonstrations in Leipzig, East-Germany, 1989-91. *World Politics, 47*(1), 42-101.

Lohmann, S. (2000). Collective action cascades: An informational rationale for the power in numbers. *Journal of Economic Surveys, 14*(5), 655-684.

Loo, A., & B.Y. Yung. (1997). *Pictorial Chinese sayings: A tale to tell (part II)* (A. Lam, Trans. Vol. 7). Hong Kong: Silk Road Pr.

López-Pérez, R. (2006). Introducing social norms in game theory. *Working Paper Institute for Empirical Research in Economics, University of Zurich, 292*, 1-26.

Luce, R.D., & H. Raiffa. (1957). *Games and decisions: Introduction and critical survey*. New York: Wiley.

Luethi, R. (2004). Software: An economic perspective and coping with high memory load in Linux (pp. 1-153): University of Zurich Master Thesis in Computer Science and Business Administration.

Luethi, R. (2005). High cost to low cost and back? The threat of regulation to low cost software production. *University of Zurich Working Paper*, 1-12.

Luethi, R. (2006, 30 June - 1 July). *Measuring FOSS project success.* Paper presented at the Ph.D. seminar of the Institute for Organization and Administrative Science, University of Zurich, Maennedorf.

Luhmann, N. (1969). *Legitimation durch Verfahren*. Neuwied: Luchterhand.

Luhmann, N. (1987). *Rechtssoziologie* (3 ed.). Opladen: Westdeutscher Verlag.

Luschen, G. (1970). Cooperation, association, and contest. *Journal of Conflict Resolution, 14*(1), 21-34.

Luthiger, B. (2006). *Spass und Software Entwicklung: Zur Motivation von Open Source Programmierern.* Unpublished Electronical Dissertation, University of Zurich, Zurich.

Lyden, F.J. (1975). Using Parsons functional analysis in study of public organizations. *Administrative Science Quarterly, 20*(1), 59-70.

Lynex, A., & P.J. Layzell. (1998). Organisational considerations for software reuse. *Annals of Software Engineering, 5*(0), 105-124.

Lyons, D. (2005). *Linux rules supercomputers*. Forbes (15. March). Retrieved 23. April, 2007, from http://www.forbes.com/home/enterprisetech/2005/03/15/cz_dl_0315linux.html.

MacKenzie, S.B., P.M. Podsakoff, & C.B. Jarvis. (2005). The problem of measurement model misspecification in behavioral and organizational research and some recommended solutions. *Journal of Applied Psychology, 90*(4), 710-730.

Macy, M.W. (1991). Chains of cooperation: Threshold effects in collective action. *American Sociological Review, 56*(6), 730-747.

Majchrzak, A., C. Wagner, & D. Yates. (2006). Corporate wiki users: Results of a survey, *International Symposium On Wikis* (21.-23. August ed., pp. 99-104). Odense: ACM Press.

Mambo Foundation. (2007). *Mambo Foundation forums* (8. February). Retrieved 8. February, 2007, from http://forum.mambo-foundation.org/.

Maner, J.K., C.L. Luce, S.L. Neuberg, R.B. Cialdini, S. Brown, & B.J. Sagarin. (2002). The effects of perspective taking on motivations for helping: Still no evidence for altruism. *Personality and Social Psychology Bulletin, 28*(11), 1601-1610.

Mann, R.J. (2006). Commercializing open source software: Do property rights still matter? *Harvard Journal of Law & Technology, 20*(1), 1-47.

Marascuilo, L.A., & J.R. Levin. (1970). Appropriate post hoc comparisons for interaction and nested hypotheses in analysis of variance designs: The elimination of type IV errors. *American Educational Research Journal, 7*(3), 397-421.

March, J.G. (1981). Footnotes on organizational change. *Administrative Science Quarterly, 26*(4), 563-577.

March, J.G. (1994). *A primer on decision making.* New York: Free Press.

March, J.G., L.S. Sproull, & M. Tamuz. (1991). Learning from samples of one or fewer. *Organization Science, 2*(1), 1-13.

March, J.G., & R.I. Sutton. (1997). Organizational performance as a dependent variable. *Organization Science, 8*(6), 698-706.

Marcuse, L. (1972). *Philosophie des Glücks von Hiob bis Freud.* Zürich: Diogenes.

Margolis, H. (1984). *Selfishness, altruism, and rationality: A theory of social choice*. Chicago: University of Chicago Press.

Margolis, H. (1990). Equilibrium norms. *Ethics, 100*(4), 821-837.

Margolis, H. (2004). Cognition and extended (nsnx) rational choice: Some early results. *Harris School Working Paper Series, 05.03*, 1-20.

Margolis, H. (2005). Cognitive anomalies and experimental economics. *Harris School Working Paper Series, 05.02*, 1-33.

Margolis, H. (2006). Cognition and extended rational choice. *Harris School Working Paper Series, 06.05B*, 1-122.

Margolis, J. (1955). A comment on the pure theory of public expenditure. *Review of Economics and Statistics, 37*(4), 347-349.

Maritz, J.S. (1995). *Distribution free statistical methods* (2 ed.). London: Chapman & Hall.

Marsh, C. (1985). Back on the bandwagon: The effect of opinion polls on public-opinion. *British Journal of Political Science, 15*(1), 51-74.

Marshall, A. (1910). *Principles of economics: An introductory volume* (6th ed.). London: Macmillan & Co.

Marwell, G., & P. Oliver. (1993). *The critical mass in collective action*. Cambridge: Cambridge University Press.

Marx, K. (1919). *Das Kapital: Kritik der politischen Ökonomie* (8 ed. Vol. 1). Hamburg: Meissner.

Masters, G.N. (1988). Item discrimination: When more is worse. *Journal of Educational Measurement, 25*(1), 15-29.

Masurel, E., & K. van Montfort. (2006). Life cycle characteristics of small professional service firms. *Journal of Small Business Management, 44*(3), 461-473.

Maxwell, S.E., & L.V. Jones. (1976). Female and male admission to graduate school: An illustrative inquiry. *Journal of Educational Statistics, 1*(1), 1-37.

Mayer, M. (2006). *Entrepreneurial Thought Leaders Seminar: Marissa Mayer (video)*. Stanford University. Retrieved 8. January, 2007, from http://stanford-online.stanford.edu/courses/msande472/060517-msande472-300.asx.

Mazen, A.M., L.A. Graf, C.E. Kellogg, & M. Hemmasi. (1987). Statistical power in contemporary management research. *Academy of Management Journal, 30*(2), 369-380.

Mcallister, I., & D.T. Studlar. (1991). Bandwagon, underdog, or projection: Opinion polls and electoral choice in Britain, 1979-1987. *Journal of Politics, 53*(3), 720-741.

McBride, D. (2007). *Letter from Darl McBride to SCO customers and partners*. SCO Group (12. January). Retrieved 25. January, 2007, from http://www.sco.com/images/company/January%2012%202007%20Letter%20from%20Darl%20McBride%20Final.pdf.

McClintock, C.G. (1972). Social motivation: A set of propositions. *Behavioral Science, 17*(5), 438-454.

McConnell, S. (1996). *Rapid development taming wild software schedules*. Redmond: Microsoft Press.

McDaniel, W.C., & F. Sistrunk. (1991). Management dilemmas and decisions: Impact of framing and anticipated responses. *Journal of Conflict Resolution, 35*(1), 21-42.

McGowan, D. (2005). Legal aspects of free and open source software. In: J. Feller, B. Fitzgerald, S. Hissam & K. R. Lakhani (Eds.), *Perspectives on free and open source software* (pp. 361-391). Cambridge: MIT Press.

McIlroy, M.D. (1991). Unix on my mind, *Virginia Computer Users Conference* (Vol. 21, pp. 1-6). Blacksburg.

McIlroy, M.D., E.N. Pinson, & B.A. Tague. (1978). Unix time sharing system forward. *Bell System Technical Journal, 57*(6), 1899-1904.

McKelvey, R.D., & T.R. Palfrey. (1992). An experimental study of the centipede game. *Econometrica, 60*(4), 803-836.

McKusick, M.K. (1999). Twenty years of Berkeley Unix: From AT&T owned to freely redistributable. In: C. DiBona, S. Ockman & M. Stone (Eds.), *Open sources: Voices from the open source revolution* (1 ed., pp. 31-46). Sebastopol: O'Reilly.

Medoff, J.L., & K.G. Abraham. (1980). Experience, performance, and earnings. *Quarterly Journal of Economics, 95*(4), 703-736.

Meier, S. (2004). *An economic analysis of pro-social behavior decisions to contribute money and time to public goods*. Zürich.

Mellström, C., & M. Johannesson. (2005). Crowding out in blood donation: Was Titmuss right? *Working Papers in Economics from Göteborg University, 180*, 1-27.

Menard, C. (2004). The economics of hybrid organizations. *Journal of Institutional and Theoretical Economics-Zeitschrift Fur Die Gesamte Staatswissenschaft, 160*(3), 345-376.

Merton, R. (1979). Foreword. In: E. Garfield (Ed.), *Citation indexing its theory and application in science, technology, and humanities* (pp. vii-xi). New York: Wiley.

Merton, R.K. (1948). The self-fulfilling prophecy. *Antioch Review, 8*(2), 193-210.

Merton, R.K. (1968a). Matthew effect in science. *Science, 159*(3810), 56-63.

Merton, R.K. (1968b). *Social theory and social structure*. New York: Free Press.

Merton, R.K. (1973). *The sociology of science: Theoretical and empirical investigations*. Chicago: University of Chicago Press.

Merton, R.K. (1993). *On the shoulders of giants: A shandean postscript*. Chicago: University of Chicago Press.

Messerschmitt, D.G., & C. Szyperski. (2003). *Software ecosystem understanding an indispensable technology and industry*. Cambridge: MIT Press.

Messick, D.M. (2000). Context, norms, and cooperation in modern society: A postscript. In: M. Van Vugt (Ed.), *Cooperation in modern society: Promoting the welfare of communities, states, and organizations* (Vol. 1, pp. 231-240). London: Routledge.

Messick, D.M., & M.B. Brewer. (1983). Solving social dilemmas: A review. In: L. Wheeler (Ed.), *Review of Personality and Social Psychology* (Vol. 4, pp. 11-44). Beverly Hills: Sage.

Messick, D.M., & C.L. Mcclelland. (1983). Social traps and temporal traps. *Personality and Social Psychology Bulletin, 9*(1), 105-110.

Meyer, C. (2005). Foreword. In: R. Goldman & R. P. Gabriel (Eds.), *Innovation happens elsewhere: Open source as business strategy* (pp. xv-xviii). Amsterdam: Elsevier.

Michelson, A.A., & E.W. Morley. (1887). On the relative motion of the earth and the luminiferous ether. *American Journal of Science, 34*(203), 333-345.

Microsoft. (2005a). *Microsoft and open source*. Retrieved 12. July, 2006, from http://www.microsoft.com/resources/sharedsource/Articles/MicrosoftandOpenSource.mspx.

Microsoft. (2005b). *Shared Source Initiative frequently asked questions*. Retrieved 12. July, 2006, from http://www.microsoft.com/resources/sharedsource/Initiative/FAQ.mspx.

Microsoft. (2005c). *Shared Source Licenses*. Retrieved 12. July, 2006, from http://www.microsoft.com/resources/sharedsource/licensingbasics/sharedsourcelicenses.mspx.

Milavsky, J.R. (1985). Early calls of election results and exit polls: Pros, cons, and constitutional considerations - Introduction. *Public Opinion Quarterly, 49*(1), 1-1.

Milgrom, P., & J. Roberts. (1992). *Economics, organization and management*. Englewood Cliffs: Prentice-Hall.

Mili, H., F. Mili, & A. Mili. (1995). Reusing software: Issues and research directions. *IEEE Transactions on Software Engineering, 21*(6), 528-562.

Mill, J.S. (1879). *Utilitarianism*. London: Longmans, Green, & Co.

Miller, A.C. (1936). Whence and whither in the gold standard? *Proceedings of the Academy of Political Science, 17*(1), 83-93.

Miller, D., & P.H. Friesen. (1983a). Strategy making and environment: The third link. *Strategic Management Journal, 4*(3), 221-235.

Miller, D., & P.H. Friesen. (1983b). Successful and unsuccessful phases of the corporate life-cycle. *Organization Studies, 4*(4), 339-356.

Miller, D., & P.H. Friesen. (1984). A longitudinal-study of the corporate life-cycle. *Management Science, 30*(10), 1161-1183.

Miller, G. (1992). *Managerial dilemmas: The political economy of hierarchy*. Cambridge: Cambridge University Press.

Mintzberg, H., & A. Mchugh. (1985). Strategy formation in an adhocracy. *Administrative Science Quarterly, 30*(2), 160-197.

Mitchell, T.R., & D. Daniels. (2003). Motivation. In: W. C. Borman, D. R. Ilgen & R. J. Klimoski (Eds.), *Industrial and organizational psychology* (Vol. 12, pp. 225-254). New York: Wiley.

Mitchell, T.R., & A.E. Mickel. (1999). The meaning of money: An individual-difference perspective. *Academy of Management Review, 24*(3), 568-578.

MITRE. (2003). Use of free and open source software (FOSS) in the U.S. Department of Defense *MITRE Report MP 02 W0000101*, 1-168.

Mitroff, I.I., & T.R. Featheringham. (1974). On systemic problem solving and error of the third kind. *Behavioral Science, 19*(6), 383-393.

Mitsubishi Research Institute. (2004a). *FLOSS-ASIA results*. Retrieved 6. September, 2006, from http://oss.mri.co.jp/floss-asia/summary_en.html.

Mitsubishi Research Institute. (2004b). *FLOSS-JP results*. Retrieved 6. September, 2006, from http://oss.mri.co.jp/floss-jp/report_en.html.

Mitsubishi Research Institute. (2004c). *Free/libre/open source software Asian developers online survey (FLOSS-ASIA)*. Retrieved 6. September, 2006, from http://oss.mri.co.jp/floss-asia/qa_en.html.

Mitsubishi Research Institute. (2004d). *Free/libre/open source software Japanese developers online survey (FLOSS-JP)*. Retrieved 6. September, 2006, from http://oss.mri.co.jp/floss-jp/qa_en.html.

Moch, M., & E. Seashore. (1981). How norms affect behaviors in and of corporations. In: P. C. Nystrom & W. H. Starbuck (Eds.), *Handbook of organizational design* (Vol. 1, pp. 210-237). Oxford: Oxford University Press.

Mockus, A., R.T. Fielding, & J.D. Herbsleb. (2002). Two case studies of open source software development: Apache and Mozilla. *ACM Transactions on Software Engineering and Methodology, 11*(3), 309-346.

Moglen, E. (1999). Anarchism triumphant: Free software and the death of copyright. *First Monday, 4*(8).

Moglen, E. (2006). *Enforcing the GNU GPL*. Free Software Foundation. Retrieved 13. June, 2006, from http://www.gnu.org/philosophy/enforcing-gpl.html.

Moody, G. (1997). The greatest OS that (n)ever was. *Wired, 5*(08).

Moon, H., L.V. Dyne, & K. Wrobel. (2005). The circumplex model and the future of organizational citizenship behavior research. In: D. L. Turnipseed (Ed.), *Handbook of organizational citizenship behavior: A review of "good soldier" activity in organizations* (pp. 3-23). New York: Nova Science Publishers.

Moores, K., & S. Yuen. (2001). Management accounting systems and organizational configuration: A life-cycle perspective. *Accounting Organizations and Society, 26*(4-5), 351-389.

Morgan, J.H. (1916). *Leaves from a field note book*. London: Macmillan & Co.

Morgenstern, O. (1948). Demand theory reconsidered. *The Quarterly Journal of Economics, 62*(2), 165-201.

Morris, D. (1991). *Warum hat das Zebra Streifen? Körpersprache und Verhaltensformen der Tiere*. München: Heyne.

Morrison, P.D., J.H. Roberts, & E. von Hippel. (2000). Determinants of user innovation and innovation sharing in a local market. *Management Science, 46*(12), 1513-1527.

Mosteller, F. (1948). A k-sample slippage test for an extreme population. *Annals of Mathematical Statistics, 19*(1), 58-65.

Mozilla Foundation. (2005). *Visual identity guidelines: Firefox*. Retrieved 12. February, 2007, from http://www.mozilla.org/foundation/identity-guidelines/firefox.html.

Mozilla Foundation. (2006a). *History: Timelines*. Retrieved 10. October, 2006, from http://wiki.mozilla.org/History:Timelines.

Mozilla Foundation. (2006b). *Mozilla trademark policy* (1.0). Retrieved 12. February, 2007, from http://www.mozilla.org/foundation/trademarks/policy.html.

Mozilla Foundation. (2007). *Annotated Mozilla Public License: Version 1.1*. Retrieved 26. June, 2006, from http://www.mozilla.org/MPL/MPL-1.1-annotated.html.

Mueller, D.C. (1989). *Public choice II*. Cambridge: Cambridge University Press.

Mueller, D.C. (2003). *Public choice III*. Cambridge: Cambridge University Press.

Mulder, L.B., E. van Dijk, H.A.M. Wilke, & D. De Cremer. (2005). The effect of feedback on support for a sanctioning system in a social dilemma: The difference between installing and maintaining the sanction. *Journal of Economic Psychology, 26*(3), 443-458.

Mundie, C. (2001). *The commercial software model: Speech transcript (The New York University Stern School of Business)*. Microsoft (3. May). Retrieved 13. March, 2007, from http://www.microsoft.com/presspass/exec/craig/05-03sharedsource.mspx.

Musée du Louvre. (2007). *Collection overview: Thirty-five thousand works of art* Retrieved 28. February, 2007, from http://www.louvre.fr/llv/oeuvres/alaune.jsp?bmLocale=en.

Myers, R.A., & B. Worm. (2003). Rapid worldwide depletion of predatory fish communities. *Nature, 423*(6937), 280-283.

Nachbar, J.H. (1998). The last word on Giffen goods? *Economic Theory, 11*(2), 403-412.

Nadeau, R., E. Cloutier, & J.H. Guay. (1993). New evidence about the existence of a bandwagon effect in the opinion formation process. *International Political Science Review, 14*(2), 203-213.

Narduzzo, A., & A. Rossi. (2003). Modularity in action: GNU/Linux and free/open source software development model unleashed. *Working Paper University Trento, 78* (27. May), 1-38.

NASA Earth Science Data System Reuse Working Group. (2006). *Initial survey findings posted*. Retrieved 13. September, 2006, from http://softwarereuse.gsfc.nasa.gov/reusenews/surveyweb.

Nasar, S., & D. Gruber. (2006). *Manifold destiny*. The New Yorker (28. August). Retrieved 13. June, 2007, from http://www.newyorker.com/archive/2006/08/28/060828fa_fact2?printable=true.

National Council Member. (2007). Report from the Swiss Parliament. *Personal communication,* 4. January.

Nature. (2001). Human genomes, public and private. *Nature, 409*(6822), 745-745.

Naval Historical Center. (1999). *The first "computer bug"*. Naval Historical Center. Retrieved 06. June, 2006, from http://www.history.navy.mil/photos/images/h96000/h96566kc.htm.

Navazio, R. (1977). Experimental approach to bandwagon research. *Public Opinion Quarterly, 41*(2), 217-225.

Nemiroff, R., & J. Bonnell. (2008). *Astronomy picture of the day: What is Hanny's Voorwerp?* NASA (25. June). Retrieved 6. August, 2008, from http://antwrp.gsfc.nasa.gov/apod/ap080625.html.

Netcraft. (2006a). *Apache now the leader in SSL servers*. Netcraft. Retrieved 06. July, 2006, from http://news.netcraft.com/archives/2006/04/26/apache_now_the_leader_in_ssl_servers.html.

Netcraft. (2006b). *April 2006 web server survey*. Netcraft. Retrieved 06. July, 2006, from http://news.netcraft.com/archives/2006/04/06/april_2006_web_server_survey.html.

Netcraft. (2007). *Netcraft web server survey: What is the market share of the different servers?* Netcraft. Retrieved 22. January, 2007, from http://survey.netcraft.com/Reports/.

Netcraft. (2008). *Netcraft web server survey: What is the market share of the different servers?* Netcraft. Retrieved 31. August, 2008, from http://survey.netcraft.com/Reports/.

Netscape. (1998). Netscape announces mozilla.org, a dedicated team and web site supporting development of free client source code. *Netscape Communications Corporation Press Release, 23. February 1998.*

Network Working Group, & S. Hambridge. (1995). *Netiquette guidelines (RFC 1855)*. Internet Engineering Task Force. Retrieved 18. April, 2006, from http://www.ietf.org/rfc/rfc1855.txt.

Network Working Group, & G. Malkin. (1993). *Internet users' glossary (RFC 1392)*. Internet Engineering Task Force. Retrieved 18. April, 2006, from http://www.ietf.org/rfc/rfc1392.txt.

Network Working Group, & G. Malkin. (1996). *Internet users' glossary (RFC 1983)*. Internet Engineering Task Force. Retrieved 18. April, 2006, from http://www.ietf.org/rfc/rfc1983.txt.

Next-Gen Podcast. (2006). *Next-Gen.Biz Podcast episode 6* (17. October). Retrieved 11. June, 2007, from http://podcast.next-gen.biz/?p=10.

Neyman, J., & E.S. Pearson. (1967). The testing of statistical hypotheses in relation to probabilities a priori. In: J. Neyman & E. S. Pearson (Eds.), *Joint statistical papers* (pp. 186-202). Berkeley: University of California Press.

Niehans, J. (2003). Transaction costs. In: J. Eatwell, M. Milgate & P. Newman (Eds.), *The new Palgrave: A dictionary of economics* (Vol. 4, pp. 676-679). Hampshire: Palgrave Macmillan Publisher.

Noelle-Neumann, E. (1974). Spiral of silence: A theory of public opinion. *Journal of Communication, 24*(2), 43-51.

Noelle-Neumann, E. (1993). *The spiral of silence: Public opinion - our social skin* (2 ed.). Chicago: University of Chicago Press.

North, D.C. (1997). *Institutions, institutional change, and economic performance*. Cambridge: Cambridge University Press.

Novell. (2005). Allianz Arena in München: Linux sorgt für glänzenden Auftritt. *ISIS Linux Report, 2*, 214.

Novick, M.R. (1966). The axioms and principal results of classical test theory. *Journal of Mathematical Psychology, 3*(1), 1-18.

Nowak, M., & K. Sigmund. (1993). A strategy of win-stay, lose-shift that outperforms tit-for-tat in the prisoner's dilemma game. *Nature, 364*(6432), 56-58.

Ntoumanis, N. (2005). A prospective study of participation in optional school physical education using a self-determination theory framework. *Journal of Educational Psychology, 97*(3), 444-453.

Nuseibeh, B. (1997). Ariane 5: Who dunnit? *IEEE Software, 14*(3), 15-16.

Nyborg, K., & M. Rege. (2003). Does public policy crowd out private contributions to public goods? *Public Choice, 115*(3), 397-418.

Nye, J.S. (2006). *Mikhail Gorbachev and the end of the cold war*. Webdiary (29. March). Retrieved 13. June, 2007, from http://webdiary.com.au/cms/?q=node/1358/print.

Nystedt, D., & IDG News Service. (2007). *One million OLPC laptop orders confirmed*. Computerworld (16. February). Retrieved 28. February, 2007, from http://www.computerworld.com.au/index.php?id=9652413.

O'Mahony, S. (2003). Guarding the commons: How community managed software projects protect their work. *Research Policy, 32*(7), 1179-1198.

O'Mahony, S. (2005). Nonprofit foundations and their role in community firm software collaboration. In: J. Feller, B. Fitzgerald, S. Hissam & K. R. Lakhani (Eds.), *Perspectives on free and open source software* (pp. 393-413). Cambridge: MIT Press.

O'Reilly, T. (1999). Lessons from open source software development. *Communications of the ACM, 42*(4), 33-37.

O'Reilly, T. (2005). *What Is Web 2.0: Design patterns and business models for the next generation of software*. Retrieved 3. December, 2006, from http://www.oreillynet.com/lpt/a/6228.

Oatley, K., & J.M. Jenkins. (1996). *Understanding emotions*. Cambridge: Blackwell Publishers.

Obstfeld, M. (1996). Models of currency crises with self-fulfilling features. *European Economic Review, 40*(3-5), 1037-1047.

OECD. (2004). *Patents and innovation: Trends and policy challenges*. OECD Publications. Retrieved 12. February, 2007, from http://www.oecd.org/dataoecd/48/12/24508541.pdf.

OECD. (2006). *Compendium of patent statistics*. OECD Publications. Retrieved 12. February, 2007, from http://www.oecd.org/dataoecd/5/19/37569377.pdf.

OECD. (2007). *Education at a glance 2007: Indicator A2 - How many students finish secondary education?* OECD. Retrieved 7. September, 2008, from http://www.oecd.org/document/30/0,3343,en_2649_39263238_39251550_1_1_1_1,00.html.

OLCP. (2007). *Concept images*. Retrieved 28. February, 2007, from http://www.laptop.org/download.en_US.html.

Olejnik, S., & J. Algina. (2000). Measures of effect size for comparative studies: Applications, interpretations, and limitations. *Contemporary Educational Psychology, 25*(3), 241-286.

Oliver, P. (1980). Rewards and punishments as selective incentives for collective action. *American Journal of Sociology, 85*(6), 1356-1375.

Oliver, P., G. Marwell, & R. Teixeira. (1985). A theory of the critical mass I: Interdependence, group heterogeneity, and the production of collective action. *American Journal of Sociology, 91*(3), 522-556.

Oliver, P.E., & G. Marwell. (1988). The paradox of group size in collective action: A theory of the critical mass II. *American Sociological Review, 53*(1), 1-8.

Oliver, P.E., & G. Marwell. (2001). Whatever happened to critical mass theory? A retrospective and assessment. *Sociological Theory, 19*(3), 292-311.

Olson, D.W., & L.E. Jasinski. (1986). Keyboard efficiency. *Byte, 11*(2), 241-244.

Olson, M. (1995). *The logic of collective action.* Cambridge: Harvard University Press.

One Laptop per Child. (2006a). *Hardware specification.* One Laptop per Child. Retrieved 26. May, 2006, from http://wiki.laptop.org/index.php?title=Hardware_specification&oldid=3270.

One Laptop per Child. (2006b). *OLPC FAQ.* One Laptop per Child. Retrieved 26. May, 2006, from http://wiki.laptop.org/index.php?title=OLPC_FAQ&oldid=3596.

OneStat.com. (2006). *Global usage share Mozilla Firefox has increased according to OneStat.com* (9 July 2006). Retrieved 13. September, 2006, from http://www.onestat.com/html/aboutus_pressbox44-mozilla-firefox-has-slightly-increased.html.

Oosterhof, A.C. (1976). Similarity of various item discrimination indexes. *Journal of Educational Measurement, 13*(2), 145-150.

Open Invention Network. (2006). *Open invention network*. Retrieved 12. February, 2007, from http://www.openinventionnetwork.com/.

Opp, K.D. (1991). GDR 89: Some notes on the causes of a spontaneous revolution. *Kölner Zeitschrift für Soziologie und Sozialpsychologie, 43*(2), 302-321.

Oram, A. (2007). *Why do people write free documentation? Results of a survey*. O'Reilly (14. June). Retrieved 24. July, 2007, from http://www.onlamp.com/pub/a/onlamp/2007/06/14/why-do-people-write-free-documentation-results-of-a-survey.html.

Orbell, J., & R.M. Dawes. (1991). A cognitive miser theory of cooperators advantage. *American Political Science Review, 85*(2), 515-528.

Orbell, J.M., & R.M. Dawes. (1993). Social welfare, cooperators advantage, and the option of not playing the game. *American Sociological Review, 58*(6), 787-800.

Orbell, J.M., & B.M. Rutherfo. (1973). Can Leviathan make life of man less solitary, poor, nasty, brutish and short. *British Journal of Political Science, 3*(4), 383-407.

Orbell, J.M., A.J.C. Vandekragt, & R.M. Dawes. (1988). Explaining discussion induced cooperation. *Journal of Personality and Social Psychology, 54*(5), 811-819.

Organ, D.W. (1977). A reappraisal and reinterpretation of the satisfaction-cause-performance hypothesis. *Academy of Management Review, 2*(1), 46-53.

Ortmann, A., & G. Gigerenzer. (1997). Reasoning in economics and psychology: Why social context matters. *Journal of Institutional and Theoretical Economics-Zeitschrift Fur Die Gesamte Staatswissenschaft, 153*(4), 700-710.

Orwell, G. (1949). *1984: A novel*. London: Secker & Warburg.

OSDL. (2005). *Press release: OSDL launches online patent commons reference library*. The Linux Foundation (15. November). Retrieved 12. February, 2007, from http://old.linux-foundation.org/newsroom/press_releases/2005/2005_11_15_beaverton.html.

OSI. (2006a). *License index*. Open Source Initiative. Retrieved 15. June, 2006, from http://www.opensource.org/licenses/.

OSI. (2006b). *The open source definition*. Open Source Initiative. Retrieved 15. June, 2006, from http://opensource.org/docs/definition.php.

Osterloh, M. (2005). Human resources management and knowledge creation. *CREMA Working Paper, 2005-09*, 1-21.

Osterloh, M., & B.S. Frey. (2000). Motivation, knowledge transfer, and organizational forms. *Organization Science, 11*(5), 538-550.

Osterloh, M., B.S. Frey, & J. Frost. (2001). Managing motivation, organization and governance. *Journal of Management and Governance, 5*(3-4), 231-239.

Osterloh, M., & S. Rota. (2007). Open source software development: Just another case of collective invention? *Research Policy, 36*(2), 157-171.

Osterloh, M.F., Bruno S. (2000). Motivation, knowledge transfer and organizational form. *Organization Science, 11*(5), 538-550.

Ostrom, E. (1997). *Governing the commons: The evolution of institutions for collective action* (Reprint ed.). Cambridge: Cambridge University Press.

Ostrom, E. (1998). A behavioral approach to the rational choice theory of collective action. *American Political Science Review, 92*(1), 1-22.

Ostrom, E. (2000a). Collective action and the evolution of social norms. *Journal of Economic Perspectives, 14*(3), 137-158.

Ostrom, E. (2000b). Crowding out citizenship. *Scandinavian Political Studies, 23*(1), 3-16.

Ostrom, E., J. Walker, & R. Gardner. (1992). Covenants with and without a sword: Self governance is possible. *American Political Science Review, 86*(2), 404-417.

Ouchi, W.G. (1980). Markets, bureaucracies and clans. *Administrative Science Quarterly, 28*(1), 129-141.

Ouimet, M.J. (2003). *The rise and fall of the Brezhnev doctrine in Soviet foreign policy*. Chapel Hill: University of North Carolina Press.

Packard, D., D. Kirby, & K.R. Lewis. (1996). *The HP way: How Bill Hewlett and I built our company*. New York: B&T.

Pareto, V. (1909). *Manuel d'économie politique* (A. Bonnet, Trans.). Paris: V. Giard & E. Brière.

Parnas, D.L. (1972). On the criteria to be used in decomposing systems into modules. *Communications of the ACM, 15*(12), 1053-1058.

Parsons, T. (1968). *The structure of social action: A study in social theory with special reference to a group of recent European writers*. New York: The Free Press.

Patalong, F. (2006). *Browsermarkt: Nach Feierabend liegt Firefox vorn*. Spiegel Online. Retrieved 7. December, 2006, from http://www.spiegel.de/netzwelt/tech/0,1518,452899,00.html.

Patent Commons Project. (2007a). *Commitments*. Retrieved 12. February, 2007, from http://www.patentcommons.org/resources/about_commitments.php.

Patent Commons Project. (2007b). *The commons*. Retrieved 12. February, 2007, from http://www.patentcommons.org/commons/.

Paulson, J.W., G. Succi, & A. Eberlein. (2004). An empirical study of open-source and closed-source software products. *IEEE Transactions on Software Engineering, 30*(4), 246-256.

Pearce, J.A., & E.C. Ravlin. (1987). The design and activation of self-regulating work groups. *Human Relations, 40*(11), 751-782.

Pearson, C.A.L. (1992). Autonomous workgroups: An evaluation at an industrial site. *Human Relations, 45*(9), 905-936.

Pearson, E.S. (1939). "Student" as statistician. *Biometrika, 30*(3/4), 210-2050.

Peddibhotla, N.B., & M.R. Subramani. (2007). Contributing to public document repositories: A critical mass theory perspective. *Organization Studies, 28*(3), 327-346.

Pedhazur, E.J. (1997). *Multiple regression in behavioral research: Explanation and prediction* (3 ed.). Fort Worth: Harcourt Brace.

Perens, B. (1999a). It's time to talk about free software again. *debian-devel mailing list, 17. February*.

Perens, B. (1999b). The open source definition. In: C. DiBona, S. Ockman & M. Stone (Eds.), *Open sources: Voices from the open source revolution* (1 ed., pp. 171-188). Sebastopol: O'Reilly.

Perry, D.E., N.A. Staudenmayer, & L.G. Votta. (1994). People, organizations, and process improvement. *IEEE Software, 11*(4), 36-45.

Persky, J. (1995). The ethology of homo economicus. *Journal of Economic Perspectives, 9*(2), 221-231.

Pettigrew, A.M. (1972). Information control as a power resource. *Sociology, 6*(2), 187-204.

Pettit, P., & R. Sugden. (1989). The backward induction paradox. *Journal of Philosophy, 86*(4), 169-182.

Pfaff, S. (2006). *Exit-voice dynamics and the collapse of East Germany: The crisis of Leninism and the revolution of 1989*. Durham: Duke University Press.

Pfaff, S., & H. Kim. (2003). Exit-voice dynamics in collective action: An analysis of emigration and protest in the East German revolution. *American Journal of Sociology, 109*(2), 401-444.

Picot, A., & H. Dietl. (1990). Transaktionskostentheorie. *Wirtschaftswissenschaftliches Studium, 19*(4), 178-184.

Pillutla, M.M., & X.P. Chen. (1999). Social norms and cooperation in social dilemmas: The effects of context and feedback. *Organizational Behavior and Human Decision Processes, 78*(2), 81-103.

Pipes, R. (1996). *A concise history of the Russian revolution*. New York: Random House.

PITAC. (2003). Report to the President: Developing open source software to advance high end computing (pp. 1-14): President's Information Technology Advisory Committee.

PledgeBank. (2006). *Frequently asked questions*. Pledge Bank. Retrieved 26. Mai, 2006, from http://www.pledgebank.com/faq.

Pluck, R.A. (1969). Review of toward a social report. *Journal of the Royal Statistical Society (Series A, General), 132*(4), 581-584.

Plutarch. (1878). *Plutarch's morals* (Vol. 3). Boston: Little, Brown, & Company.

Polacheck, T. (2006). Tuna longline catch rates in the Indian Ocean: Did industrial fishing result in a 90% rapid decline in the abundance of large predatory species? *Marine Policy, 30*(5), 470-482.

Polanyi, M. (2000). The republic of science: Its political and economic theory. *Minerva, 38*(1), 1-21.

Poppe, M. (2005). The specificity of social dilemma situations. *Journal of Economic Psychology, 26*(3), 431-441.

Poppe, U., R. Eckert, & I.-S. Kowalczuk (Eds.). (1995). *Zwischen Selbstbehauptung und Anpassung: Formen des Widerstandes und der Opposition in der DDR*. Berlin: Ch.Links.

Popper, K.R. (1984). *Logik der Forschung* (8 ed.). Tübingen: J.C.B. Mohr Paul Siebeck.

Posner, E.A. (2000). *Law and social norms*. Cambridge: Harvard University Press.

Posner, R.A. (1997). Social norms and the law: An economic approach. *American Economic Review, 87*(2), 365-369.

Post, T., M.J. Van den Assem, G. Baltussen, & R. Thaler. (2007). Deal or no deal? Decision making under risk in a large payoff game show. *SSRN Working Paper Series, 636505*(June).

Poulin, J.S. (1997). *Measuring software reuse principles, practices, and economic models*. Reading: Addison-Wesley.

Poulsen, K. (2004). *Software bug contributed to blackout*. Security Focus. Retrieved 12. June, 2006, from http://www.securityfocus.com/news/8016.

Prahalad, C.K., & G. Hamel. (1990). The core competence of the corporation. *Harvard Business Review, 68*(3), 79-91.

Pratten, C.F. (1980). The manufacture of pins. *Journal of Economic Literature, 18*(1), 93-96.

Project Gutenberg. (2006). *General FAQ*. Retrieved 5. October, 2006, from http://www.gutenberg.org/wiki/Gutenberg:General_FAQ.

Pruitt, D.G., & J.Z. Rubin. (1986). *Social conflict: Escalation, stalemate, and settlement* (1 ed.). New York: Random House.

Pryor, F.L. (1971). Empirical note on tipping point. *Land Economics, 47*(4), 413-417.

Pugh, D.S., & D.J. Hickson. (1976). *Organizational behaviour in its context: The Aston programme I*. Farnborough: Saxon House.

Pugh, D.S., & C.R. Hinings. (1976). *Organizational structure extensions and replications: The Aston programme II*. Farnborough: Saxon House.

Pugh, D.S., & R. Payne. (1977). *Organizational behaviour in its context: The Aston programme III*. Farnborough: Saxon House.

Quinn, R.E., & K. Cameron. (1983). Organizational life cycles and shifting criteria of effectiveness: Some preliminary evidence. *Management Science, 29*(1), 33-51.

Rabin, M. (1993). Incorporating fairness into game theory and economics. *American Economic Review, 83*(5), 1281-1302.

Raiffa, H. (1961). Risk, Ambiguity, and the Savage Axioms - Comment. *Quarterly Journal of Economics, 75*(4), 690-699.

Raiffa, H. (1968). *Decision analysis: Introductory lectures on choices under uncertainty*. Reading: Addison-Wesley.

Randerson, J. (2006). *Meet the cleverest man in the world (who's going to say no to a $1m prize)*. The Guardian (16. August). Retrieved 13. June, 2007, from http://www.guardian.co.uk/frontpage/story/0,,1851095,00.html.

Rapoport, A. (1985). Provision of public goods and the MCs experimental paradigm. *American Political Science Review, 79*(1), 148-155.

Rapoport, A., G. Bornstein, & I. Erev. (1989). Intergroup Competition for Public-Goods - Effects of Unequal Resources and Relative Group-Size. *Journal of Personality and Social Psychology, 56*(5), 748-756.

Rapoport, A., D.V. Budescu, R. Suleiman, & E. Weg. (1992). Social dilemmas with uniformly distributed resources. In: W. B. G. Liebrand, D. M. Messick & H. A. M. Wilke (Eds.), *Social dilemmas: Theoretical issues and research findings* (pp. 43-57). Oxford: Pergamon Press.

Rasch, G. (1980). *Probabilistic models for some intelligence and attainment tests*. Chicago: University of Chicago Press.

Rasmusen, E. (1989). *Games and information: An introduction to game theory*. Oxford: Blackwell.

Raven, B.H., & J.R.P. French. (1958). Legitimate power, coercive power, and observability in social influence. *Sociometry, 21*(2), 83-97.

Raymond, E.S. (1998). Homesteading the noosphere. *First Monday, 3*(10).

Raymond, E.S. (1999a). *The cathedral and the bazaar: Musings on Linux and open source by an accidental revolutionary* (1 ed.). Cambridge: O'Reilly.

Raymond, E.S. (1999b). *Shut up and show them the code*. Retrieved 12. July, 2006, from http://www.catb.org/~esr/writings/shut-up-and-show-them.html.

Raymond, E.S. (2002). Foreword. In: J. Feller & B. Fitzgerald (Eds.), *Understanding open source software development*. London: Addison-Wesley.

Raymond, E.S. (2003). *The jargon file (version 4.4.7)*. Retrieved 19. June, 2006, from http://www.catb.org/~esr/jargon/html/go01.html.

Raymond, E.S. (2004). *The art of Unix programming*. Boston: Addison-Wesley.

Reel, J.S. (1999). Critical success factors in software projects. *IEEE Software, 16*(3), 18-+.

Reeve, J., & E.L. Deci. (1996). Elements of the competitive situation that affect intrinsic motivation. *Personality and Social Psychology Bulletin, 22*(1), 24-33.

Republik Österreich. (1999). *44. Bundesgesetz über die Gewinnung von Blut und Blutbestandteilen in Blutspendeeinrichtungen*. Österreichisches Rotes Kreuz (10. March). Retrieved 17. June, 2007, from http://www.roteskreuz.at/61_body.html.

Rheinberg, F. (2006). Intrinsische Motivation und Flow-Erleben. In: J. Heckhausen (Ed.), *Motivation und Handeln* (3 ed., pp. 331-354). Berlin: Springer.

Ring, P.S., & A.H. Vandeven. (1992). Structuring cooperative relationships between organizations. *Strategic Management Journal, 13*(7), 483-498.

Ritchie, D.M. (1978). *The UNIX Time-sharing System: A Retrospective*. Retrieved 22. August, 2006, from http://cm.bell-labs.com/cm/cs/who/dmr/retro.html.

Ritchie, D.M. (1984). Reflections on software research. *Communications of the ACM, 27*(8), 758-760.

Ritchie, D.M. (1996). *The evolution of the Unix time-sharing system*. Retrieved 8. March, 2007, from http://cm.bell-labs.com/cm/cs/who/dmr/hist.html.

Ritchie, D.M. (2001). *Space Travel: Exploring the solar system and the PDP-7*. Retrieved 22. August, 2006, from http://cm.bell-labs.com/cm/cs/who/dmr/spacetravel.html.

Rivest, R.L., A. Shamir, & L. Adleman. (1978). Method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM, 21*(2), 120-126.

Robbins, J. (2005). Adopting open source software engineering (OSSE) practices by adapting OSSE tools. In: J. Feller, B. Fitzgerald, S. Hissam & K. R. Lakhani (Eds.), *Perspectives on free and open source software* (pp. 245-264). Cambridge: MIT Press.

Roberts, H.V. (1963). Risk, Ambiguity, and the Savage Axioms: Comment. *Quarterly Journal of Economics, 77*(2), 327-336.

Roberts, J.A., I.H. Hann, & S.A. Slaughter. (2006). Understanding the motivations, participation, and performance of open source software developers: A longitudinal study of the Apache projects. *Management Science, 52*(7), 984-999.

Robinson, W.S. (1950). Ecological correlations and the behavior of individuals. *American Sociological Review, 15*(3), 351-357.

Robles, G. (2004). A Software Engineering approach to Libre Software. In: R. Büttner, Y. Gasmi, R. A. Gehring, A. John, S. Kharitoniouk, B. Lutterbeck, P. Stewin & M. Unger (Eds.), *Open Source Jahrbuch 2004: Zwischen Softwareentwicklung und Gesellschaftsmodell* (pp. 193.208). Berlin: Lehmanns Media.

Robles, G., H. Schneider, I. Tretkowski, & N. Weber. (2001a). *Who is doing it?* , Technical University of Berlin, Berlin.

Robles, G., H. Schneider, I. Tretkowski, & N. Weber. (2001b). *WIDI survey*. Technical University of Berlin. Retrieved 7. February, 2007, from http://widi.berlios.de/index.php3.

Roch, S.G., & C.D. Samuelson. (1997). Effects of environmental uncertainty and social value orientation in resource dilemmas. *Organizational Behavior and Human Decision Processes, 70*(3), 221-235.

Rocha, H.O., & S. Ghoshal. (2006). Beyond self-interest revisited. *Journal of Management Studies, 43*(3), 585-619.

Rogozinski, J. (2000). *Honor among thieves: Captain Kidd, Henry Every, and the pirate democracy in the Indian Ocean*. Mechanicsburg: Stackpole Books.

Rohlfs, J.H. (2001). *Bandwagon effects in high-technology industries*. Cambridge: MIT Press.

Rohner, D., A. Winestein, & B.S. Frey. (2006). Ich bin auch ein Lemming: Herding and consumption capital in arts and culture. *Working Paper Institute for Empirical Research in Economics, University of Zurich, 270*, 1-19.

Ronson, J. (2006). *Cold sweat*. The Guardian (28. January). Retrieved 8. January, 2007, from http://www.guardian.co.uk/weekend/story/0,,1695278,00.html.

Rose, C. (1986). The comedy of the commons: Custom, commerce, and inherently public property. *University of Chicago Law Review, 53*(3), 711-781.

Rosen, S. (1999). Potato paradoxes. *Journal of Political Economy, 107*(6), S294-S313.

Ross, D. (2006). *Game theory*. Stanford Encyclopedia of Philosophy. Retrieved 25. June, 2007, from http://plato.stanford.edu/archives/spr2006/entries/game-theory/.

Ross, W.D. (1958). Commentary. In: Aristotle & W. D. Ross (Eds.), *Aristotle's metaphysics* (4 ed., Vol. 2, pp. 159-500). London: Oxford University Press.

Rossi, P.H., W.A. Sampson, C.E. Bose, G. Jasso, & J. Passel. (1974). Measuring household social standing. *Social Science Research, 3*(3), 169-190.

Rost, J. (2004). *Lehrbuch Testtheorie - Testkonstruktion* (2 ed.). Bern: Huber.

Rousseau, J.-J. (1761). *A discourse upon the origin and foundation of the inequality among mankind*. London: R. & J. Dodsley.

Rousseau, J.-J. (1911). *Emile, or, treatise on education* (B. Foxley, Trans.). London: J.M. Dent & Sons Ltd.; E.P. Dutton & Co.

Rowe, R. (2007). *DreamWorks animation "Shrek the third": Linux feeds an ogre*. Linux Journal (5. June). Retrieved 18. June, 2007, from http://www.linuxjournal.com/article/9653.

Royce, W. (1987). Managing the development of large software systems: Concepts and techniques. *Proceedings of the 9th International Conference on Software Engineering,* 328-338.

Ruffin, R.J. (1972). Pollution in a Crusoe economy. *Canadian Journal of Economics, 5*(1), 110-118.

Rutherford, M.W., P.F. Buller, & P.R. McMullen. (2003). Human resource management problems over the life cycle of small to medium-sized. *Human Resource Management, 42*(4), 321-335.

Rutte, C., & M. Taborsky. (2007). Generalized reciprocity in rats. *PLOS Biology, 5*(7), 1421-1425.

Ryan, R.M., & E.L. Deci. (2000a). Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary Educational Psychology, 25*(1), 54-67.

Ryan, R.M., & E.L. Deci. (2000b). Self-determination theory and the facilitating of intrinsic motivation, social development, and well-being. *American Psychologist, 55*(1), 68-78.

Sackett, P.R., & C.J. DeVore. (2001). Counterproductive behaviors at work. In: N. Anderson, D. S. Ones, H. K. Sinangil & C. Viswesvaran (Eds.), *Handbook of industrial, work and organizational psychology* (Vol. 1, pp. 145-164). London: Sage.

Sahr, M. (2002). *Zeit für Märchen: Kreativer und medienorientierter Umgang mit einer epischen Kurzform*. Baltmannsweiler: Schneider-Verlag Hohengehren.

Salovey, P., J.D. Mayer, & D.L. Rosenhan. (1991). Mood and helping: Mood as a motivator of helping and helping as a regulator of mood. In: M. S. Clark (Ed.), *Review of personality and social psychology* (pp. 215-237). New York: Sage.

Samardzija, M.R. (2007). The obvious war. *Science, 315*(5809), 190-191.

Sametinger, J. (1997). *Software engineering with reusable components*. Berlin: Springer.

Samuelson, C.D. (1991). Perceived task difficulty, causal attributions, and preferences for structural change in resource dilemmas. *Personality and Social Psychology Bulletin, 17*(2), 181-187.

Samuelson, C.D., & D.M. Messick. (1986). Inequities in access to and use of shared resources in social dilemmas. *Journal of Personality and Social Psychology, 51*(5), 960-967.

Samuelson, P.A. (1954). The pure theory of public expenditure. *Review of Economics and Statistics, 36*(4), 387-390.

Samuelson, P.A. (1967). *Economics: an introductory analysis* (7 ed.). New York: McGraw-Hill.

Sanchez, R., & A. Heene. (1997). A competence perspective on strategic learning and knowledge management. In: R. Sanchez & A. Heene (Eds.), *Strategic learning and knowledge management* (pp. 3-15). Chichester: Wiley.

Sanchez, R., & J.T. Mahoney. (1996). Modularity, flexibility, and knowledge management in product and organization design. *Strategic Management Journal, 17*(Winter Special Issue), 63-76.

Sansone, C., & J.M. Harackiewicz. (2000). Looking beyond rewards: The problem and promise of intrinsic motivation. In: C. Sansone & J. M. Harackiewicz (Eds.), *Intrinsic and extrinsic motivation: The search for optimal motivation and performance* (pp. 1-9). San Diego: Academic Press.

Santos, F.C., M.D. Santos, & J.M. Pacheco. (2008). Social diversity promotes the emergence of cooperation in public goods games. *Nature, 454*, 213-216.

Sauley, K.S., & A.G. Bedeian. (1989). .05: A case of the tail wagging the distribution. *Journal of Management, 15*(2), 335-344.

Savage, R.P. (1994). The paradox of nontransitive dice. *American Mathematical Monthly, 101*(5), 429-436.

Schade, J. (2003). The psychology of the crowd and the Monday Demonstrations of Leipzig 1989. *Forum der Psychoanalyse, 19*(1), 42-48.

Schaller, M., & R.B. Cialdini. (1990). Happiness, sadness, and helping: A motivational integration. In: E. T. Higgins & R. M. Sorrentino (Eds.), *Handbook of motivation and cognition* (Vol. 2, pp. 265-298). New York: Guilford Press.

Scharfstein, D.S., & J.C. Stein. (1990). Herd behavior and investment. *American Economic Review, 80*(3), 465-479.

Schawlow, A. (1982). Going for the gaps. *Stanford Magazine, Fall*, 42.

Schei, V. (2004, 15. July 2004). *Cat among the pigeons? Cooperators' response to individualists in negotiating groups.* Paper presented at the IACM 17th Annual Conference Pittsburgh.

Schelling, T.C. (1960). *The strategy of conflict*. Cambridge: Harvard University Press.

Schelling, T.C. (1971a). Dynamic models of segregation. *Journal of Mathematical Sociology, 1*(2), 143-186.

Schelling, T.C. (1971b). Ecology of micromotives. *Public Interest, 25*(Fall), 61-98.

Schelling, T.C. (1972). A process of residential segregation: Neighborhood tipping. In: A. H. Pascal & Rand Corporation. (Eds.), *Racial discrimination in economic life* (pp. 157-184). Lexington: Lexington Books.

Schelling, T.C. (1973a). Hockey helmets, concealed weapons, and daylight saving: Study of binary choices with externalities. *Journal of Conflict Resolution, 17*(3), 381-428.

Schelling, T.C. (1973b). A process of residential segregation: neighborhood tipping. In: A. Pascal (Ed.), *Racial discrimination in economic life* (pp. 157-184). Lexington: D. C. Heath.

Schelling, T.C. (1978). *Micromotives and macrobehavior* (1 ed.). New York: Norton.

Schelling, T.C. (1980a). The intimate contest for self-command. *Public Interest, 60*(Summer), 94-118.

Schelling, T.C. (1980b). *The strategy of conflict*. Cambridge: Harvard University Press.

Schelling, T.C. (1985). Enforcing rules on oneself. *Journal of Law, Economics and Organization, 1*(2), 357-374.

Schermerhorn, J.R., J.G. Hunt, & R.N. Osborn. (2005). *Organizational behavior* (9 ed.). New York: Wiley.

Scheufele, D.A., & P. Moy. (2000). Twenty-five years of the spiral of silence: A conceptual review and empirical outlook. *International Journal of Public Opinion Research, 12*(1), 3-28.

Schiff, J. (1985). Does government spending crowd out charitable contributions? *National Tax Journal, 38*(4), 535-546.

Schmid, R.E. (2006). *Rare and costly stamps to go on display*. Washington Post (25. May). Retrieved 28. March, 2007, from http://www.washingtonpost.com/wp-dyn/content/article/2006/05/25/AR2006052501464.html.

Schmidt, A.U. (2006). Multi-level markets and incentives for information goods. *Information Economics and Policy, 18*(2), 125-138.

Schneider, K. (1996). Intrinsisch (autotelisch) motiviertes Verhalten dargestellt an den Beispielen des Neugierverhaltens sowie verwandter Verhaltenssysteme (Spielen und leistungsmotiviertes Handeln). In: J. Kuhl & H. Heckhausen (Eds.), *Motivation, Volition und Handlung* (pp. 119-152). Göttingen: Hogrefe.

Schneider, S.K., & G.V. Northcraft. (1999). Three social dilemmas of workforce diversity in organizations: A social identity perspective. *Human Relations, 52*(11), 1445-1467.

Schneier, B. (1996). *Applied cryptography: Protocols, algorithms, and source code in C* (2 ed.). New York: Wiley.

Schneier, B. (2004). *Secrets and lies: Digital security in a networked world*. New York: Wiley Computer Publishing.

Schneier, B. (2006). *Schneier on Security: Prison shivs*. Retrieved 11. December, 2006, from http://www.schneier.com/blog/archives/2006/08/prison_shivs.html.

Schnell, R. (1997). *Nonresponse in Bevölkerungsumfragen: Ausmass, Entwicklung und Ursachen*. Opladen: Leske & Budrich.

Schram, A. (2005). Artificiality: The tension between internal and external validity in economic experiments. *Journal of Economic Methodology, 12*(2), 225-237.

Schuessler, R. (1990). Threshold effects and the decline of cooperation. *Journal of Conflict Resolution, 34*(3), 476-494.

Schumpeter, J.A. (1939a). *Business cycles: A theoretical, historical, and statistical analysis of the capitalist process* (3 ed. Vol. 2). New York: McGraw-Hill.

Schumpeter, J.A. (1939b). *Business cycles: A theoretical, historical, and statistical analysis of the capitalist process* (3 ed. Vol. 1). New York: McGraw-Hill.

Schwartz-Shea, P., & R.T. Simmons. (1995). Social dilemmas and perceptions: Experiments on framing and inconsequentiality. In: D. A. Schroeder (Ed.), *Social dilemmas: Perspectives on individuals and groups* (pp. 87-103). Westport: Praeger.

Schwartz, S.H. (1994). Beyond individualism/collectivism: New cultural dimensions of values. In: U. Kim, H. C. Triandis, C. Kagitcibasi, S.-C. Choi & G. Yoon (Eds.), *Individualism and collectivism: Theory, method, and applications* (pp. 85-119). Thousand Oaks: Sage.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics, 6*(2), 461-464.

Schweik, C.M., & A. Semenov. (2003). The institutional design of open source programming: Implications for addressing complex public policy and management problems. *First Monday, 5*(1).

Schweizerische Eidgenossenschaft. (1994). *Bundesgesetz über die Krankenversicherung (832.10)*. Retrieved 17. Juni, 2007, from http://www.admin.ch/ch/d/sr/832_10/index.html.

Schweizerische Eidgenossenschaft. (2006). *Obligationenrecht: Artikel 321d* (4. July). Retrieved 14. March, 2007, from http://www.admin.ch/ch/d/sr/220/a321d.html.

Scott, B.R. (1971). Stages of corporate development. *Harvard Business School Note, Case No. 9-371-294.*

Sedlmeier, P., & G. Gigerenzer. (1989). Do studies of statistical power have an effect on the power of studies. *Psychological Bulletin, 105*(2), 309-316.

Seife, C. (2003). Columbia disaster underscores the risky nature of risk analysis. *Science, 299*(5609), 1001-1002.

Sen, A.K. (1977). Rational fools: A critique of the behavioural foundations of economic theory. *Philosophy and Public Affairs, 6*(4), 317-344.

Sen, A.K. (1997). *On economic inequality* (enl. ed.). Oxford: Oxford University Press.

Sethi, R., & E. Somanathan. (2003). Understanding reciprocity. *Journal of Economic Behavior & Organization, 50*(1), 1-27.

Settele, C. (2006, 27. January). Die verborgenen Perlen des Browsers. *Neue Zürcher Zeitung,* p. 61.

Shakespeare, W. (1992a). *Hamlet*. New York: Washington Square Press.

Shakespeare, W. (1992b). *Macbeth*. New York: Washington Square Press.

Shankland, S. (2002). *IBM: Linux investment nearly recouped*. CNET News.com (29. January). Retrieved 25. June, 2007, from http://news.com.com/2100-1001-825723.html.

Shankland, S. (2006). *HP offers Debian Linux support*. CNET News.com (15. August). Retrieved 3. January, 2007, from http://www.zdnetasia.com/news/software/0,39044164,39390316,00.htm.

Shapira, Z. (1976). Expectancy determinants of intrinsically motivated behavior. *Journal of Personality and Social Psychology, 34*(6), 1235-1244.

Shapiro, C. (2001). Navigating the patent thicket: Cross licenses, patent pools, and standard-setting. *SSRN Working Paper Series, 273550*, 1-32.

Shapiro, C., & H.R. Varian. (1999). *Information rules: A strategic guide to the network economy*. Boston: Harvard Business School Press.

Sheldon, K.M., & A.J. Elliot. (1999). Goal striving, need satisfaction, and longitudinal well-being: The self-concordance model. *Journal of Personality and Social Psychology, 76*(3), 482-497.

Sheldon, K.M., A.J. Elliot, Y. Kim, & T. Kasser. (2001). What is satisfying about satisfying events? Testing 10 candidate psychological needs. *Journal of Personality and Social Psychology, 80*(2), 325-339.

Sherry, F. (1986). *Raiders and rebels: The golden age of piracy* (1 ed.). New York: Hearst Marine Books.

Shortridge, R.T., & M. Myring. (2004). Defining principles based accounting standards. *CPA Journal, 74*(8), 34-37.

Shroer, J., & G. Hertel. (2007). Voluntary engagement in an open web-based encyclopedia: Wikipedians, and why they do it. *University of Wuerzburg Working Paper, 8. January*, 1-49.

Shubik, M. (1971). Dollar auction game: Paradox in noncooperative behavior and escalation. *Journal of Conflict Resolution, 15*(1), 109-&.

Sievers, H.-J. (1990). *Stundenbuch einer deutschen Revolution: Die Leipziger Kirchen im Oktober 1989*. Zollikon: G2W-Verlag.

Sim, S.M., & R.I. Rasiah. (2006). Relationship between item difficulty and discrimination indices in true/falsetype multiple choice questions of a para-clinical multidisciplinary paper. *Annals Academy of Medicine Singapore, 35*(2), 67-71.

Siminoff, L.A., A. Zhang, N. Colabianchi, C.M.S. Sturm, & Q.S. Shen. (2000). Factors that predict the referral of breast cancer patients onto clinical trials by their surgeons and medical oncologists. *Journal of Clinical Oncology, 18*(6), 1203-1211.

Simmel, G. (1964). *The sociology of Georg Simmel* (K. H. Wolff, Trans.). New York: Free Press.

Simon, H.A. (1954). Bandwagon and underdog effects and the possibility of election predictions. *Public Opinion Quarterly, 18*(3), 245-253.

Simon, H.A. (1981). *The Sciences of the artificial* (2 ed.). Cambridge: MIT Press.

Simon, H.A. (1991). Organization and markets. *Journal of Economic Perspectives, 5*(2), 25-44.

Simpson, E.H. (1951). The interpretation of interaction in contingency tables. *Journal of the Royal Statistical Society Series B-Statistical Methodology, 13*(2), 238-241.

Singh, G. (2006). A shift from significance test to hypothesis test through power analysis in medical research. *Journal of Postgraduate Medicine, 52*(2), 148-150.

Singh, S., & J. Lynch. (1998). *Fermat's enigma: The epic quest to solve the world's greatest mathematical problem*. New York: Anchor Books.

Sink, E. (2006). *Why we all sell code with bugs*. The Guardian (25. May). Retrieved 25. June, 2007, from http://technology.guardian.co.uk/online/insideit/story/0,,1782071,00.html.

Skalaban, A. (1988). Do the polls affect elections? Some 1980 evidence. *Political Behavior, 10*(2), 136-150.

Skewes, S. (1933). On the difference $\pi(x) - Li(x)$. *Journal of the London Mathematical Society, 8*, 277-283.

Skyrms, B. (2003). *The stag hunt and the evolution of social structure*. Cambridge: Cambridge University Press.

Slashdot. (2006). *Debian delayed by disenchanted developers* (20. December). Retrieved 3. January, 2007, from http://linux.slashdot.org/article.pl?sid=06/12/20/191215.

Slattery, O., R. Lu, J. Zheng, F. Byers, & X. Tang. (2004). Stability comparison of recordable optical discs: A study of error rates in harsh conditions. *Journal of Research of the National Institute of Standards and Technology, 109*(5), 517-524.

Smith, A. (1976). *An inquiry into the nature and causes of the wealth of nations*. Chicago: University of Chicago Press.

Smith, A. (2000). *The theory of moral sentiments*. Amherst: Prometheus Books.

Smith, C.A., D.W. Organ, & J.P. Near. (1983). Organizational citizenship behavior: Its nature and antecedents. *Journal of Applied Psychology, 68*(4), 653-663.

Smith, L., & P. Sorensen. (2000). Pathological outcomes of observational learning. *Econometrica, 68*(2), 371-398.

Smith, T.W. (1983). The hidden 25 percent: An analysis of nonresponse on the 1980 General Social Survey. *Public Opinion Quarterly, 47*(3), 386-404.

Snidal, D. (1979). Public goods, property rights, and political organizations. *International Studies Quarterly, 23*(4), 532-566.

Sniezek, J.A., D.R. May, & J.E. Sawyer. (1990). Social uncertainty and interdependence: A study of resource allocation decisions in groups. *Organizational Behavior and Human Decision Processes, 46*(2), 155-180.

Snow, D.A., S.A. Soule, & H. Kriesi. (2004). *The Blackwell companion to social movements*. Malden: Blackwell.

Solé, R.V., & J. Bascompte. (2006). *Self-organization in complex ecosystems*. Princeton: Princeton University Press.

Solézenicyn, A.I. (2005). *Der Archipel Gulag* (A. Peturnig & E. Walter, Trans. Vol. 1). Reinbek: Rowohlt.

Solheim, S. (2006). *Linux branching out, says IBM exec*. InfoWorld (8. March). Retrieved 25. June, 2007, from http://www.infoworld.com/article/06/03/08/76234_HNlinuxbranching_1.html.

Solow, R.M. (1971). Blood and thunder. *Yale Law Journal, 80*(8), 1696-1711.

Somaya, D. (2003). Strategic determinants of decisions not to settle patent litigation. *Strategic Management Journal, 24*(1), 17-38.

Sonnemans, J., A. Schram, & T. Offerman. (1998). Public good provision and public bad prevention: The effect of framing. *Journal of Economic Behavior & Organization, 34*(1), 143-161.

Sontag, R. (2006). *Ein tolles Wochenende ist vorbei ...* Chemnitzer Linux-Tage. Retrieved 10. April, 2006, from http://chemnitzer.linux-tage.de/2006/presse/news/lt2006-mail7.html.

SourceForge. (2004a). *Most active projects: Last week* (21. May). Retrieved 21. May, 2004, from http://sourceforge.net/top/mostactive.php?type=week and http://sourceforge.net/top/mostactive.php?type=week&offset=50.

SourceForge. (2004b). *Software map* (17. May). Retrieved 17. May, 2004, from http://sourceforge.net/softwaremap/trove_list.php?form_cat=XXX (replace XXX with numbers 13 to 631).

SourceForge. (2006a). *Registered projects and registered users* (15. June). Retrieved 15. June, 2006, from http://sourceforge.net/index.php.

SourceForge. (2006b). *Software map* (15. June). Retrieved 15. June, 2006, from http://sourceforge.net/softwaremap/trove_list.php?form_cat=XXX (replace XXX with numbers 13 to 631).

SourceForge. (2006c). *Software search* (11. July). Retrieved 11. July, 2006, from http://sourceforge.net/search/.

SourceForge. (2007). *Most active projects: Last week* (20. February). Retrieved 20. February, 2007, from http://sourceforge.net/top/mostactive.php?type=week and http://sourceforge.net/top/mostactive.php?type=week&offset=50.

Spear, S., & H.K. Bowen. (1999). Decoding the DNA of the Toyota production system. *Harvard Business Review, 77*(5), 96-106.

Spector, P.E. (2001). Research methods in industrial and organizational psychology. In: N. Anderson, D. S. Ones, H. K. Sinangil & C. Viswesvaran (Eds.), *Handbook of industrial, work and organizational psychology* (Vol. 1, pp. 10-26). London: Sage.

Spence, M. (1973). Job market signaling. *Quarterly Journal of Economics, 87*(3), 355-374.

Spender, J.C. (1996). Making knowledge the basis of a dynamic theory of the firm. *Strategic Management Journal, 17*(Winter Special Issue), 45-62.

Spender, J.C., & R.M. Grant. (1996). Knowledge and the firm: Overview. *Strategic Management Journal, 17*(Winter Special Issue), 5-9.

Sproull, L., & S. Kiesler. (1986). Reducing social context cues: Electronic mail in organizational communication. *Management Science, 32*(11), 1492-1512.

SPSS. (2005). *SPSS base 14.0 user's guide*. Chicago: Prentice Hall.

St. Laurent, A.M. (2004). *Understanding open source & free software licensing* (1 ed.). Sebastopol: O'Reilly Media Inc.

Stallman, R. (1983). *Initial announcement*. Free Software Foundation. Retrieved 13. June, 2006, from http://www.gnu.org/gnu/initial-announcement.html.

Stallman, R. (1999). *Re: Fake award to "the collaborative software community"*. debian-devel mailing list (15. February). Retrieved 25. June, 2007, from http://lists.debian.org/debian-devel/1999/02/msg01474.html.

Stallman, R. (2001a). *The GNU project*. Free Software Foundation. Retrieved 13. June, 2006, from http://www.gnu.org/gnu/thegnuproject.html.

Stallman, R. (2001b). *Why "free software" is better than "open source"*. Free Software Foundation. Retrieved 13. June, 2006, from http://www.gnu.org/philosophy/free-software-for-freedom.html.

Stallman, R. (2002). *Linux and the GNU project*. Free Software Foundation. Retrieved 13. June, 2006, from http://www.gnu.org/gnu/linux-and-gnu.html.

Stallman, R. (2004a). *The free software community after 20 Years: With great but incomplete success, what now?* Free Software Foundation. Retrieved 13. June, 2006, from http://www.gnu.org/philosophy/use-free-software.html.

Stallman, R. (2004b). *The rise of the free software movement: Freedom from proprietary control: An Interview with Richard Stallman*. Multinational Monitor (Volume 25, Nr. 7/8). Retrieved 2. February, 2007, from http://multinationalmonitor.org/mm2004/07012004/july-aug04interviewstall.html.

Stallman, R. (2005). *On the Netscape Public License*. Retrieved 10. October, 2006, from http://www.gnu.org/philosophy/netscape-npl.html.

Stallman, R. (2006). *Fighting software patents: Singly and together* (25. October). Retrieved 12. February, 2007, from http://www.gnu.org/philosophy/fighting-software-patents.html.

Stecklow, S. (2005). *The $100 laptop moves closer to reality*. Wall Street Journal Online (14. November 2005). Retrieved 26. May, 2006, from http://online.wsj.com/public/article/SB113193305149696140-442o71jo_IlBrLpyUeeOdsqDs7E_20061113.html?mod=tff_main_tff_top.

Stevens, S.S. (1946). On the theory of scales of measurement. *Science, 103*(2684), 677-680.

Stevens, S.S. (1951). Mathematics, measurement and psychophysics. In: S. S. Stevens (Ed.), *Handbook of experimental psychology* (pp. 1-46). New York: Wiley.

Stevenson, C. (1995). *Software engineering productivity a practical guide*. London: Chapman & Hall.

Stevenson, G.G. (1991). *Common property economics: A general theory and land use applications*. Cambridge: Cambridge University Press.

Stigler, G.J. (1947). Notes on the history of the Giffen paradox. *Journal of Political Economy, 55*(2), 152-156.

Stigler, G.J. (1987). *The theory of price* (4 ed.). New York: Collier Macmillan.

Stouten, J., D. De Cremer, & E. van Dijk. (2006). Violating equality in social dilemmas: Emotional and retributive reactions as a function of trust, attribution, and honesty. *Personality and Social Psychology Bulletin, 32*(7), 894-906.

Straffin, P.D. (1980). The prisoner's dilemma. *UMAP Journal, 1*, 102-103.

Straus, S.G. (1997). Technology, group process, and group outcomes: Testing the connections in computer-mediated and face-to-face groups. *Human-Computer Interaction, 12*(3), 227-266.

Stuckhoff, R. (2005). *Interview: IBM's Adam Jollans outlines Linux strategy*. WebSphere Journal (19. November). Retrieved 25. June, 2007, from http://websphere.sys-con.com/read/148249.htm.

Subversion. (2006). *Bite-sized coding tasks*. CollabNet. Retrieved 15. March, 2007, from http://subversion.tigris.org/project_tasks.html#bite-sized.

Sudman, S. (1986). Do exit polls influence voting behavior? *Public Opinion Quarterly, 50*(3), 331-339.

Sun. (2007). *Free and open source Java*. Retrieved 3. July, 2007, from http://www.sun.com/software/opensource/java/index.jsp.

Suntharalingam, G., M.R. Perry, S. Ward, S.J. Brett, A. Castello-Cortes, M.D. Brunner, et al. (2006). Cytokine storm in a phase 1 trial of the anti-CD28 monoclonal antibody TGN1412. *New England Journal of Medicine, 355*(10), 1018-1028.

Swann, G.M.P. (2002). The functional form of network effects. *Information Economics and Policy, 14*(3), 417-429.

Swatz, A. (2006). *Who writes Wikipedia?* Retrieved 5. September, 2006, from http://www.aaronsw.com/weblog/whowriteswikipedia.

Swedish Institute. (2003). *Fact sheets on Sweden: The health care system in Sweden* (No. FS 76 y Vpb): Swedish Institute.

Sweet, D. (2001). *KDE 2.0 development*. Indianapolis: Sams.

Swisher, K. (2004, 29. July). 'Wiki' may alter how employees work together. *Wall Street Journal,* p. B1.

Swiss-American Chamber of Commerce. (1995). *Swiss Code of Obligations: English translation of the official text* (Vol. 1). Zürich: Swiss-American Chamber of Commerce.

Swiss Parliamentary Service. (2007). *Cura Vista: Database of the Swiss Parliament*. Retrieved 8. January, 2007, from http://www.parlament.ch/e/homepage/su-suchen/su-curia-vista.htm.

Taeuber, K.E. (1970). Toward a social report: Review article. *Journal of Human Resources, 5*(3), 354-360.

Takeuchi, K. (1976). Distribution of informational statistics and a criterion of model fitting. *Suri-Kagaku, 153,* 12-18.

Tanenbaum, A.S. (2000). *MINIX license change*. comp.os.minix (7. April). Retrieved 27. November, 2006, from http://groups.google.de/group/comp.os.minix/browse_thread/thread/b41bfd3c16622f2b/d12a0dde04b7f232#d12a0dde04b7f232.

Tanzanian Parks Authority. *Elephant shrew and buffalo weaver: Elephant shrew*. Retrieved 19. August, 2008, from http://www.serengeti.org/littlefive_shrew.html.

Taylor, F.W. (1911). *The principles of scientific management*. New York: Harper & Brothers.

Taylor, M. (1982). *Community, anarchy, and liberty*. Cambridge: Cambridge University Press.

Taylor, M. (1987). *The possibility of cooperation*. Cambridge: Cambridge University Press.

Tedeschi, J.T., T. Bonoma, & Lindskol.S. (1970). Threateners reactions to prior announcement of behavioral compliance or defiance. *Behavioral Science, 15*(2), 171-&.

Tenbrunsel, A.E., & D.M. Messick. (1999). Sanctioning systems, decision frames, and cooperation. *Administrative Science Quarterly, 44*(4), 684-707.

Theraulaz, G., E. Bonabeau, & J.L. Deneubourg. (1995). Self organization of hierarchies in animal societies: The case of the primitively eusocial wasp Polistes Dominulus Christ. *Journal of Theoretical Biology, 174*(3), 313-323.

Thibaut, J., & L. Walker. (1975). *Procedural justice*. Hillsdale: Erlbaum.

Thibaut, J.W., & H.H. Kelley. (1959). *The social psychology of groups*. New York: John Wiley & Sons.

Thomas, W.I., & D.S.T. Thomas. (1928). *The child in America: Behavior problems and programs*. New York,: A. A. Knopf.

Thompson, K. (1984). Reflections on trusting trust. *Communications of the ACM, 27*(8), 761-763.

Thomson. (2004). International abbreviations and acronyms database. München: Saur.

Thurston, R. (2006). *Debian delayed as community loses interest*. ZDNet UK (20. December). Retrieved 3. January, 2007, from http://www.zdnetasia.com/news/software/0,39044164,61976570,00.htm.

Thurstone, L.L. (1927). A law of comparative judgment. *Psychological Review, 34*(4), 273-286.

Thurstone, L.L. (1928). Attitudes can be measured. *American Journal of Sociology, 33*(4), 529-554.

Thurstone, L.L., & E.J. Chave. (1929). *The measurement of attitude: A psychophysical method and some experiments with a scale for measuring attitude toward the church*. Chicago: University of Chicago Press.

Tilly, C. (1984). *Big structures, large processes, huge comparisons*. New York: Russell Sage Foundation.

Tilly, C. (2004). *Social movements, 1768-2004*. Boulder: Paradigm Publishers.

Titmuss, R.M. (1970). *The gift relationship: From human blood to social policy*. London: Allen & Unwin.

Titmuss, R.M., A. Oakley, & J. Ashton. (1997). *The gift relationship: From human blood to social policy* (2 ed.). New York: New Press.

Tocqueville, A.d. (1964). *Souvenirs*. Paris: Gallimard.

Tocqueville, A.d., & J. Bonner. (1856). *The old regime and the revolution*. New York: Harper & Brothers.

Tommer, B. (2006, 29. June). Spenderblut wird knapp. *Neue Zürcher Zeitung,* p. 43.

Tönnies, F. (1912). *Gemeinschaft und Gesellschaft: Grundbegriffe der reinen Soziologie* (2 ed.). Berlin: Karl Curtius.

Toole, B.A. (1996). Ada Byron, Lady Lovelace: An analyst and metaphysician. *IEEE Annals of the History of Computing, 18*(3), 4-12.

Torbert, W.R. (1974). Pre-bureaucratic and post-bureaucratic stages of organization development. *Interpersonal Development, 5*(1), 1-25.

Torvalds, L. (1991). *What would you like to see most in minix?* Newsgroup comp.os.minix (25. August). Retrieved 25. June, 2007, from http://groups.google.com/group/comp.os.minix/msg/b813d52cbc5a044b?hl=en.

Torvalds, L. (2003). *Re: Linux GPL and binary module exception clause?* Linux kernel mailing list (3. December). Retrieved 25. June, 2007, from http://www.ussg.iu.edu/hypermail/linux/kernel/0312.0/0670.html.

Torvalds, L., & D. Diamond. (2001). *Just for fun*. London: Texere.

Torvalds, L., & R.A. Ghosh. (1998). What motivates free software developers? *First Monday, 3*(3).

Towns, A. (2006). *Response to "Position statement to the Dunc-Tanc "experiment""*. Mailing list "debian-devel-announce@lists.debian.org" (27. October). Retrieved 3. January, 2007, from http://people.debian.org/~terpstra/message/20061027.123514.5970b22d.en.html.

Townsend, R.E. (1995). Fisheries self-governance: Corporate or cooperative structures. *Marine Policy, 19*(1), 39-45.

Travica, B. (1999). *New organizational designs: Information aspects*. Stamford: Ablex Pub. Corp.

Treaster, J.B., & D. Sontag. (2005). Desperate survivors beg for help; Chaos spreads in New Orleans. *The International Herald Tribune,* p. 1.

Treffert, D.A. (1989). *Extraordinary people: Understanding "idiot savants"* (1st ed.). New York: Harper & Row.

Tucker, A.W. (1980). On jargon: The prisoner's dilemma. *UMAP Journal, 1*, 101.

Tuckman, B.W. (1965). Developmental sequence in small-groups. *Psychological Bulletin, 63*(6), 384-399.

Tuckman, B.W. (1984). Citation classic: Developmental sequence in small-groups. *Current Contents/Social & Behavioral Sciences, 34*, 14.

Tuckman, B.W., & M.A.C. Jensen. (1977). Stages of small-group development revisited. *Group & Organization Management (Group Organization Studies), 2*(4), 419-427.

Tversky, A., & D. Kahneman. (1971). Belief in law of small numbers. *Psychological Bulletin, 76*(2), 105-110.

Tversky, A., & D. Kahneman. (1981). The framing of decisions and the psychology of choice. *Science, 221*(4481), 453-458.

Tyler, T.R. (1988). What Is procedural justice: Criteria used by citizens to assess the fairness of legal procedures. *Law & Society Review, 22*(1), 103-135.

Tyler, T.R. (1990). *Why people obey the law*. New Haven: Yale University Press.

Tyler, T.R., & P. Degoey. (1995). Collective restraint in social dilemmas: Procedural justice and social identification effects on support for authorities. *Journal of Personality and Social Psychology, 69*(3), 482-497.

U.S.-Canada Power System Outage Task Force. (2004). *Final report on the August 14, 2003 blackout in the United States and Canada: Causes and recommendations*. Ottawa: Natural Resources Canada.

U.S. Census Bureau. (2004). *Current Population Survey 2003: Educational attainment in the United States*. Retrieved 14. August, 2006, from http://www.census.gov/prod/2004pubs/p20-550.pdf.

U.S. Census Bureau. (2005). *Current Population Survey 2004: Annual social and economic (ASEC) supplement*. Retrieved 14. August, 2006, from http://www.census.gov/apsd/techdoc/cps/cpsmar04.pdf.

U.S. Department of Health Education and Welfare. (1969). *Toward a social report*. Washington: U.S. Government Printing Office.

U.S. National Institutes of Health. (2005). *An Introduction to clinical trials*, from http://www.clinicaltrials.gov/ct/info/whatis#phases.

UBS. (2008). *Shareholder report on UBS's write-downs*. Zurich.

Udehn, L. (1993). 25 Years with the logic of collective action. *Acta Sociologica, 36*(3), 239-261.

UNESCO. (1997). *International standard classification of education 1997*. Retrieved 14. August, 2006, from http://www.uis.unesco.org/TEMPLATE/pdf/isced/ISCED_A.pdf.

UNESCO. (2003). *Education indicators: Technical guidelines*. Retrieved 14. August, 2006, from http://www.uis.unesco.org/ev.php?ID=5202_201&ID2=DO_TOPIC.

UNESCO. (2006). *Water: A shared responsibility. The United Nations world water development report 2*. Paris: Berghahn Books.

Unger, L.S. (1991). Altruism as a motivation to volunteer. *Journal of Economic Psychology, 12*(1), 71-100.

UniProt. (2006). *Welcome to UniProt*. Retrieved 18. September, 2006, from http://www.ebi.uniprot.org/index.shtml.

University of Mannheim, University of Tennessee, & NERSC/LBNL. (2006a). *Top500 supercomputer sites: November 2006: Operating system family*. Retrieved 22. January, 2007, from http://www.top500.org/stats/28.

University of Mannheim, University of Tennessee, & NERSC/LBNL. (2006b). *Top500 supercomputer sites: November 2006: Top500 list (Excel)*. Retrieved 22. January, 2007, from http://www.top500.org/static/lists/2006/11/TOP500_200611.xls.

UNU-MERIT, Universidad Rey Juan Carlos, University of Limerick, Society for Public Information Spaces, Business Innovation Centre of Alto Adige-Südtirol, & A. Ghosh. (2006). *Economic impact of open source software on innovation and the competitiveness of the information and communication technologies (ICT) sector in the EU*. European Commission: Directorate General of Enterprise and Industry. Retrieved 22. January, 2007, from http://ec.europa.eu/enterprise/ict/policy/doc/2006-11-20-flossimpact.pdf.

Uther, H.-J. (1990). Hans im Glück. In: K. Ranke, I. Köhler & R. W. Brednich (Eds.), *Enzyklopädie des Märchens: Gott und Teufel auf Wanderschaft - Hyltéen-Cavallius* (Vol. 6, pp. 487-494). Berlin: Walter de Gruyter.

Valentinov, V. (2007). The property rights approach to nonprofit organization: The role of intrinsic motivation. *Public Organization Review, 7*(1), 41-55.

Välimäki, M. (2005). *The rise of open source licensing: A challenge to the use of intellectual property in the software industry* (1st ed.). Helsinki: Turre.

Valoris. (2003). *Comparative assessment of open documents formats market overview*. European Commission. Retrieved 22. January, 2007, from http://ec.europa.eu/idabc/servlets/Doc?id=17982.

Valverde, S., G. Theraulaz, J. Gautrais, V. Fourcassie, & R.V. Sole. (2006). Self-organization patterns in wasp and open source communities. *IEEE Intelligent Systems, 21*(2), 36-40.

Van de Ven, A. (2001). *Binary only module overview*. Linux kernel mailing list (24. September). Retrieved 25. June, 2007, from http://www.uwsg.iu.edu/hypermail/linux/kernel/0109.3/0102.html.

Van de Ven, A.H., & R.J. Koenig. (1976). Determinants of coordination modes within organizations. *American Sociological Review, 41*(2), 322-388.

Van den Bos, K. (1999). What are we talking about when we talk about no-voice procedures? On the psychology of the fair outcome effect. *Journal of Experimental Social Psychology, 35*(6), 560-577.

Van der Heijden, E.C.M., J.H.M. Nelissen, J.J.M. Potters, & H.A. Verbon. (1998). The poverty game and the pension game: The role of reciprocity. *Journal of Economic Psychology, 19*(1), 5-41.

Van der Poel, C.L., E. Siefried, & W.P. Schaasberg. (2002). Paying for blood donations: Still a risk? *Vox Sanguinis, 83*(4), 285-293.

Van Prooijen, J.-W., K. van den Bos, & H. Wilke. (2004). The role of standing in the psychology of procedural justice: Towards theoretical integration. *European Review of Social Psychology, 15*, 33-58.

Van Vugt, M., & D. de Cremer. (1999). Leadership in social dilemmas: The effects of group identification on collective actions to provide public goods. *Journal of Personality and Social Psychology, 76*(4), 587-599.

Van Vugt, M., R.M. Meertens, & P.A.M. van Lange. (1995). Car versus public transportation: The role of social value orientations in a real life social dilemma. *Journal of Applied Social Psychology, 25*(3), 258-278.

Van Vugt, M., P.A.M. van Lange, & R.M. Meertens. (1996). Commuting by car or public transportation? A social dilemma analysis of travel mode judgments. *European Journal of Social Psychology, 26*(3), 373-395.

Vanzandt, D.E. (1993). The lessons of the lighthouse: "Government" or "private" provision of goods. *Journal of Legal Studies, 22*(1), 47-72.

Vaughan-Nichols, S.J. (2006). *Mambo executives, developers fight for project control.* eWeek.com. Retrieved 15. November, 2006, from http://www.eweek.com/article2/0,1895,1850298,00.asp.

Veblen, T.B. (1975). *The theory of the leisure class: An economic study of institutions*. New York: MacMillan.

Velleman, P.F., & L. Wilkinson. (1993). Nominal, ordinal, interval, and ratio typologies are misleading. *American Statistician, 47*(1), 65-72.

Viégas, F.B., M. Wattenberg, & K. Dave. (2004). Studying cooperation and conflict between authors with history flow visualizations. *Proceedings of the SIGCHI conference on Human factors in computing systems (Vienna, 24.-29. April), 6*(1), 575-582.

Voegelin, C.F., & F.M.R. Voegelin. (1977). *Classification and index of the world's languages*. New York: Elsevier.

von Hippel, E. (1988). *The sources of innovation.* New York: Oxford University Press.

von Hippel, E. (2005a). *Democratizing innovation.* Cambridge: MIT Press.

von Hippel, E. (2005b). Open source software projects as user innovation networks. In: J. Feller, B. Fitzgerald, S. Hissam & K. R. Lakhani (Eds.), *Perspectives on free and open source software* (pp. 267-278). Cambridge: MIT Press.

von Hippel, E., & G. von Krogh. (2003). Open source software and the "private-collective" innovation model: Issues for organization science. *Organization Science, 14*(2), 209-223.

von Krogh, G., S. Spaeth, & K.R. Lakhani. (2003). Community, joining, and specialization in open source software innovation: A case study. *Research Policy, 32*(7), 1217-1241.

von Krogh, G., & E. von Hippel. (2003). Special issue on open source software development. *Research Policy, 32*(7), 1149-1157.

von Krogh, G., & E. von Hippel. (2006). The promise of research on open source software. *Management Science, 52*(7), 975-983.

Von Neumann, J., & O. Morgenstern. (1947). *Theory of games and economic behavior* (2 ed.). Princeton: Princeton University Press.

Vose, M.G., G. William, & D. Knuth. (1986). Computer science considerations: Donald Knuth speaks on his involvement with digital typography. *Byte, 11*(2), 169-172.

Vowles, J. (2002). Did the polls influence the vote? A case study of the 1999 New Zealand general election. *Political Science, 54*(1), 67-77.

Wagenaar, W.A., K. Gideon, & S. Lichtenstein. (1997). Islanders and hostages: Deep surface structure of decision problems. In: W. M. Goldstein & R. M. Hogarth (Eds.), *Research on judgment and decision making: Currents, connections, and controversies* (pp. 552-565). Cambridge: Cambridge University Press.

Wagenaar, W.A., G. Keren, & S. Lichtenstein. (1988). Islanders and hostages: Deep and surface structures of decision problems. *Acta Psychologica, 67*(2), 175-189.

Wagenseil, P. (2008). *'I Am Rich' iPhone application retails for $1,000*. FOXNEWS.COM (6. August). Retrieved 12. August, 2008, from http://www.foxnews.com/story/0,2933,398903,00.html.

Wagnerová, A. (2006). *In Wahrheit leben: Zum 70. Geburtstag von Václav Havel - eine Erinnerung*. Neue Zürcher Zeitung (5. October). Retrieved 14. June, 2007, from http://www.nzz.ch/2006/10/05/fe/articleE9UD8.html.

Wall, J.A., & M.W. Blum. (1991). Negotiations. *Journal of Management, 17*(2), 273-303.

Wall, T.D., N.J. Kemp, P.R. Jackson, & C.W. Clegg. (1986). Outcomes of autonomous workgroups: A long-term field experiment. *Academy of Management Journal, 29*(2), 280-304.

Walters, A.E., A.F. Stuhlmacher, & L.L. Meyer. (1998). Gender and negotiator competitiveness: A meta-analysis. *Organizational Behavior and Human Decision Processes, 76*(1), 1-29.

Walton, D.N. (1989). *Informal logic: A handbook for critical argumentation*. Cambridge: Cambridge University Press.

Wang, X.T. (1996). Domain-specific rationality in human choices: Violations of utility axioms and social contexts. *Cognition, 60*(1), 31-63.

Warneken, F., B. Hare, A.P. Melis, D. Hanus, & M. Tomasello. (2007). Spontaneous altruism by chimpanzees and young children. *PLOS Biology, 5*(7), 1414-1420.

Watson, C., & L.R. Hoffman. (1996). Managers as negotiators: A test of power versus gender as predictors of feelings, behavior, and outcomes. *Leadership Quarterly, 7*(1), 63-85.

Weber, A.A. (1998). Sources of currency crisis: An empirical analysis. *CEPR Working Paper*(25).

Weber, J.M., S. Kopelman, & D.M. Messick. (2004). A conceptual review of decision making in social dilemmas: Applying a logic of appropriateness. *Personality and Social Psychology Review, 8*(3), 281-307.

Weber, M. (1990). *Wirtschaft und Gesellschaft: Grundriss der verstehenden Soziologie* (5 ed.). Tübingen: Mohr.

Weber, M. (1992). *The Protestant ethic and the spirit of capitalism*. London: Routledge.

Weber, S. (2004). *The success of open source*. Cambridge: Harvard University Press.

Weibel, A., K. Rost, & M. Osterloh. (2007). Crowding-out of intrinsic motivation: Opening the black box. *SSRN Working Paper Series, 957770*, 1-22.

Weinberg, G.M. (1971). *The psychology of computer programming*. New York: Van Nostrand Reinhold.

Weinberg, G.M. (1999). Egoless programming. *IEEE Software, 16*(1), 118-120.

Weinshtein, S.I. (2005). *Geheimnisvolles Tuwa: Expeditionen in das Herz Asiens*. Oststeinbek: Alouette Verlag.

Weinstock, C.B. (2005). Making lightning strike twice. In: J. Feller, B. Fitzgerald, S. Hissam & K. R. Lakhani (Eds.), *Perspectives on free and open source software* (pp. 143-159). Cambridge: MIT Press.

Weintraub, S. (2001). *Silent night: The story of the World War I Christmas truce*. New York: Free Press.

Weise, G. (1975). *Psychologische Leistungstests: Ein Handbuch für Studium und Praxis*. Göttingen: C. J. Hogrefe.

Weitman, S.R. (1978). Prosocial behavior and its discontents. In: L. Wispé (Ed.), *Altruism, sympathy, and helping psychological and sociological principles* (pp. 229-246). New York: Academic Press.

Wel, J. (1975). Least squares fitting of an elephant. *Chemtech, 5*(2), 128-129.

Welte, H. (2006). *Buried alive in GPL violations* (Wed, 22 Feb 2006). Retrieved 26. June, 2006, from http://gnumonks.org/%7Elaforge/weblog/2006/02/22/.

Welzel, C., R. Inglehart, & H.D. Klingemann. (2003). The theory of human development: A cross-cultural analysis. *European Journal of Political Research, 42*(3), 341-379.

Wenger, E.C., & W.M. Snyder. (2000). Communities of practice: The organizational frontier. *Harvard Business Review, 78*(1), 139-145.

Wheeler, D.A. (2005). *Why open source software / free software (OSS/FS, FLOSS, or FOSS)? Look at the numbers!* Retrieved 27. November, 2006, from http://www.dwheeler.com/oss_fs_why.html.

White, M.V. (2003). Robinson Crusoe. In: J. Eatwell, M. Milgate & P. Newman (Eds.), *The new Palgrave: A dictionary of economics* (Vol. 4, pp. 217-218). Hampshire: Palgrave Macmillan Publisher.

White, N. (1995). Conflicting parts of happiness in Aristotle ethics. *Ethics, 105*(2), 258-283.

White, R.W. (1959). Motivation reconsidered: The concept of competence. *Psychological Review, 66*(5), 297-333.

White, R.W., & R. Lippitt. (1960). Leader behavior and member reaction in three "social climates". In: D. Cartwright & A. F. Zander (Eds.), *Group dynamics: Research and theory* (2 ed., pp. 527-553). Evanston: Row Peterson & Company.

Whitlock, N. (2001). *The security implications of open source software*. IBM (1. March). Retrieved 20. December, 2007, from http://www.ibm.com/developerworks/linux/library/l-oss.html.

Whittingham, M.J., P.A. Stephens, R.B. Bradbury, & R.P. Freckleton. (2006). Why do we still use stepwise modelling in ecology and behaviour? *Journal of Animal Ecology, 75*(5), 1182-1189.

Wichman, H. (1970). Effects of isolation and communication on cooperation in a two-person game. *Journal of Personality and Social Psychology, 16*(1), 114-120.

Wicksell, K. (1896). *Finanztheoretische Untersuchungen nebst Darstellung und Kritik des Steuerwesens Schwedens*. Jena: Fischer.

Wikipedia. (2006a). *Category: Linux conferences*. Retrieved 9. January, 2006, from http://en.wikipedia.org/wiki/Category:Linux_conferences.

Wikipedia. (2006b). *Informational cascade*. Retrieved 8. January, 2007, from http://en.wikipedia.org/w/index.php?title=Informational_cascade&oldid=84548503.

Wikipedia. (2006c). *Linuxtag*. Retrieved 9. January, 2006, from http://de.wikipedia.org/wiki/Linuxtag.

Wikipedia. (2006d). *Power structure*. Retrieved 6. September, 2006, from http://meta.wikimedia.org/wiki/Power_structure.

Wikipedia. (2006e). *Three men make a tiger*. Retrieved 22. January, 2007, from http://en.wikipedia.org/w/index.php?title=Three_men_make_a_tiger&oldid=91926204.

Wikipedia. (2006f). *Wikipedia*. Retrieved 6. September, 2006, from http://en.wikipedia.org/wiki/Wikipedia.

Wikipedia. (2006g). *Wikipedia statistics: All languages*. Retrieved 6. September, 2006, from http://stats.wikimedia.org/EN/TablesWikipediaZZ.htm.

Wikipedia. (2006h). *Wikipedia statistics: Article count (official)*. Retrieved 6. September, 2006, from http://stats.wikimedia.org/EN/TablesArticlesTotal.htm.

Wikipedia. (2006i). *Wikipedia: List of Wikipedians by number of edits*. Retrieved 6. September, 2006, from http://en.wikipedia.org/wiki/Wikipedia:List_of_Wikipedians_by_number_of_edits.

Wikipedia. (2006j). *Wikipedia: Size of Wikipedia*. Retrieved 4. December, 2006, from http://en.wikipedia.org/wiki/Wikipedia:Size_of_Wikipedia.

Wikipedia. (2007a). *Deal or no deal* (7. July). Retrieved 9. July, 2007, from http://en.wikipedia.org/w/index.php?title=Deal_or_No_Deal&oldid=143146592.

Wikipedia. (2007b). *Nupedia*. Retrieved 17. January, 2007, from http://en.wikipedia.org/w/index.php?title=Nupedia&oldid=100673268.

Wikipedia. (2007c). *Usage share of web browsers*. Retrieved 15. February, 2007, from http://en.wikipedia.org/w/index.php?title=Usage_share_of_web_browsers&oldid=104833768.

Wikipedia. (2007d). *Wikipedia: Vandalism*. Retrieved 17. January, 2007, from http://en.wikipedia.org/w/index.php?title=Wikipedia:Vandalism&oldid=101253617.

Wilcox, J. (2000). *IBM to spend $1 billion on Linux in 2001*. CNET News.com (20. December). Retrieved 25. June, 2007, from http://news.com.com/2100-1001-249750.html.

Wildschut, T., B. Pinter, J.L. Vevea, C.A. Insko, & J. Schopler. (2003). Beyond the group mind: A quantitative review of the interindividual-intergroup discontinuity effect. *Psychological Bulletin, 129*(5), 698-722.

Williams, K.D., J.M. Jackson, & S.J. Karau. (1995). Collective hedonism: A social loafing analysis of social dilemmas. In: D. A. Schroeder (Ed.), *Social dilemmas: Perspectives on individuals and groups* (pp. 117-142). Westport: Praeger.

Williams, S. (2002). *Free as in freedom: Richard Stallman's crusade for free software*. Sebastopol: O'Reilly.

Williamson, O.E. (1975). *Markets and hierarchies: Analysis and antitrust Implications*. New York: Free Press.

Williamson, O.E. (1985). *The economic institutions of capitalism*. New York: Free Press.

Williamson, O.E. (1991). Comparative economic organization: The analysis of discrete structural alternatives. *Administrative Science Quarterly, 36*(2), 269-296.

Wilson, E.O. (1978). The genetic evolution of altruism. In: L. Wispé (Ed.), *Altruism, sympathy, and helping: Psychological and sociological principles* (pp. 11-37). New York: Academic Press.

Wilson, E.O. (2005). Kin selection as the key to altruism: Its rise and fall. *Social Research, 72*(1), 159-166.

Wilson, J.M., S.G. Straus, & B. McEvily. (2006). All in due time: The development of trust in computer-mediated and face-to-face teams. *Organizational Behavior and Human Decision Processes, 99*(1), 16-33.

Wilson, J.Q., & G.L. Kelling. (1982). Broken windows: The police and neighborhood safety. *Atlantic Monthly, March*, 29-39.

Winer, B.J. (1962). *Statistical principles in experimental design*. New York: McGraw-Hill.

WIPO. (2008). *Berne Convention: Contracting parties*. Retrieved 19. March 2008, from http://www.wipo.int/treaties/en/ShowResults.jsp?lang=en&treaty_id=15.

Wispé, L. (1978). *Altruism, sympathy, and helping: Psychological and sociological principles*. New York: Academic Press.

Wolf, G. (1995). *The curse of Xanadu*. Wired (3.06). Retrieved 11. June, 2007, from http://www.wired.com/wired/archive/3.06/xanadu_pr.html.

Wolfson, H.A. (1958). The plurality of immovable movers in Aristotle and Averroës. *Harvard Studies in Classical Philology, 63*, 233-253.

Wood, B.D., & A. Doan. (2003). The politics of problem definition: Applying and testing threshold models. *American Journal of Political Science, 47*(4), 640-653.

Woods, R.I. (1977). Population turnover, tipping points and Markov chains. *Transactions of the Institute of British Geographers, 2*(4), 473-489.

Woodworth, R.S. (1918). *Dynamic psychology*. New York: Columbia University Press.

Wright, P.M., B.B. Dunford, & S.A. Snell. (2001). Human resources and the resource based view of the firm. *Journal of Management, 27*(6), 701-721.

Wrong, D.H. (1961). The oversocialized conception of man in modern sociology. *American Sociological Review, 26*(2), 183-193.

Wu, C.H., R. Apweiler, A. Bairoch, D.A. Natale, W.C. Barker, B. Boeckmann, et al. (2006). The universal protein resource (UniProt): An expanding universe of protein information. *Nucleic Acids Research, 34*(Database Issue), D187-D191.

Xiaoshan, P. (2006). *Computational modeling of human and social behaviors for emergency egress analysis.* Stanford University, Stanford.

Yamagishi, T. (1986). The provision of a sanctioning system as a public good. *Journal of Personality and Social Psychology, 51*(1), 110-116.

Yamagishi, T. (1988a). Exit from the group as an individualistic solution to the free rider problem in the United States and Japan. *Journal of Experimental Social Psychology, 24*(6), 530-542.

Yamagishi, T. (1988b). The provision of a sanctioning system in the United States and Japan. *Social Psychology Quarterly, 51*(3), 265-271.

Yamagishi, T. (1988c). Seriousness of social dilemmas and the provision of a sanctioning system. *Social Psychology Quarterly, 51*(1), 32-42.

Yamagishi, T., K.S. Cook, & M. Watabe. (1998). Uncertainty, trust, and commitment formation in the United States and Japan. *American Journal of Sociology, 104*(1), 165-194.

Yamagishi, T., & M. Yamagishi. (1994). Trust and commitment in the United States and Japan. *Motivation and Emotion, 18*(2), 129-166.

Yates, F. (1934). Contingency tables involving small numbers and the chi-square test. *Supplement to the Journal of the Royal Statistical Society, 1*(2), 217-235.

Yin, C.C. (1998). Equilibria of collective action in different distributions of protest thresholds. *Public Choice, 97*(4), 535-567.

Young, H.P. (1996). The economics of convention. *Journal of Economic Perspectives, 10*(2), 105-122.

Young, R., & W.G. Rohm. (2000). *Der Red Hat Coup: Wie die Open Source-Bewegung und Red Hat die Softwareindustrie revolutionieren - und Microsoft überrumpeln*. Bonn: MITP-Verlag.

Yuan, Y., J. Fulk, M. Shumate, P.R. Monge, J.A. Bryant, & M. Matsaganis. (2005). Individual participation in organizational information commons: The impact of team level social influence and technology specific competence. *Human Communication Research, 31*(2), 212-240.

Yukl, G. (1998). *Leadership in organizations* (4 ed.). London: Prentice-Hall International.

Yule, G.U., & M.G. Kendall. (1950). *An introduction to the theory of statistics* (14 ed.). London: Charles Griffin.

ZA. (2001). *Overview of the datasets of the GDR and the new federal states*. GESIS (ZA) (29. January). Retrieved 2. April, 2007, from http://www.gesis.org/en/data_service/gdr_nfs/data/studienprofile.htm.

Zabel, R.W., C.J. Harvey, S.L. Katz, T.P. Good, & P.S. Levin. (2003). Ecologically sustainable yield: Marine conservation requires a new ecosystem based concept for fisheries management that looks beyond sustainable yield for individual fist species. *American Scientist, 91*(2), 150-157.

ZAIG. (1989a). *Hinweise auf beachtenswerte Reaktionen von Mitgliedern und Funktionären der SED zu einigen aktuellen Aspekten der Lage in der DDR und zum innerparteilichen Leben*. BStU (11. September). Retrieved 29. March, 2007, from http://www.bstu.bund.de/cln_043/nn_916548/DE/MfS-DDR-Geschichte/Revolutionskalender/August-1989/Dokumentenseiten/11-September_b/11-September_b_tabelle.html_nnn=true.

ZAIG. (1989b). *Hinweise auf die Reaktion der Bevölkerung im Zusammenhang mit der ständigen Ausreise von Bürgern aus der DDR bzw. dem ungesetzlichen Verlassen der DDR*. BStU (13. September). Retrieved 29. March, 2007, from http://www.bstu.bund.de/cln_043/nn_916548/DE/MfS-DDR-Geschichte/Revolutionskalender/August-1989/Dokumentenseiten/13-September_b/13-September-b_tabelle.html_nnn=true.

Zenger, T.R. (1994). Explaining organizational diseconomies of scale in research and development: Agency problems and the allocation of engineering talent, ideas, and effort by firm size. *Management Science, 40*(6), 708-729.

Zerbe, W.J., & D.L. Paulhus. (1987). Socially desirable responding in organizational behavior: A reconception. *Academy of Management Review, 12*(2), 250-264.

Zhu, K., K.L. Kraemer, V. Gurbaxani, & S.X. Xu. (2006). Migration to open standard interorganizational systems: Network effects, switching costs, and path dependency. *MIS Quarterly, 30*(August Special Issue), 515-539.

Zhu, Z.C. (2001). Towards an integrating programme for information systems design: An oriental case. *International Journal of Information Management, 21*(1), 69-90.

Zückert, H. (2003). *Allmende und Allmendaufhebung: Vergleichende Studien zum Spätmittelalter bis zu den Agrarreformen des 18./19. Jahrhunderts*. Stuttgart: Lucius und Lucius.

ZUMA. (2005). *National identity II: ISSP 2003 codebook* (No. ZA Study 3910). Koeln: University Koeln.

CV summary

Personal information

Name	Bernhard Kuster
Family situation	Single
Citizenship	Switzerland
Place of origin	Uznach (SG) and Eschenbach (SG)
Place of birth	Uznach (SG)
Date of birth	November 4, 1974

Education

1989 – 1994	Business college in Wattwil
1994 – 2001	University education in sociology (major), business administration (first minor), and political science (second minor) at the University of Zürich.
	Master thesis: "Situationsbedingte Entscheidungsfindung"; with an empirical study at ABB Switzerland; Reviewer: Prof. E. Ruehli
1998	Visiting student at the University of Chicago
2002 – 2009	Doctoral Studies in economics at the University of Zürich
	Doctoral thesis: "The dynamics of collective action in free and open source projects"; with an empirical study; Reviewer: Prof. M. Osterloh and Prof. E. Franck

Award

NORC (University of Chicago)	First price General Social Survey (GSS) Student Paper Competition 1999 (undergraduate division)